UNIVERSITY OF ALBERTA

ADAPTIVE PREDICTIVE PID

BY

RANDY MARVIN MILLER ©

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA
SPRING, 1995

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN IRREVOCABLE NON-EXCLUSIVE LICENCE ALLOWING THE NATIONAL LIBRARY OF CANADA TO REPRODUCE, LOAN, DISTRIBUTE OR SELL COPIES OF HIS/HER THESIS BY ANY MEANS AND IN ANY FORM OR FORMAT, MAKING THIS THESIS AVAILABLE TO INTERESTED PERSONS.

THE AUTHOR RETAINS OWNERSHIP OF THE COPYRIGHT IN HIS/HER THESIS. NEITHER THE THESIS NOR SUBSTANTIAL EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT HIS/HER PERMISSION.

L'AUTEUR A ACCORDE UNE LICENCE IRREVOCABLE ET NON EXCLUSIVE PERMETTANT A LA BIBLIOTHEQUE NATIONALE DU CANADA DE REPRODUIRE, PRETER, DISTRIBUER OU VENDRE DES COPIES DE SA THESE DE QUELQUE MANIERE ET SOUS QUELQUE FORME QUE CE SOIT POUR METTRE DES EXEMPLAIRES DE CETTE THESE A LA DISPOSITION DES PERSONNE INTERESSEES.

L'AUTEUR CONSERVE LA PROPRIETE DU DROIT D'AUTEUR QUI PROTEGE SA THESE. NI LA THESE NI DES EXTRAITS SUBSTANTIELS DE CELLE-CI NE DOIVENT ETRE IMPRIMES OU AUTREMENT REPRODUITS SANS SON AUTORISATION.

ISBN 0-612-01639-0

Canada

UNIVERSITY OF ALBERTA
LIBRARY RELEASE FORM

Name of Author:	Randy Marvin Miller
Title of Thesis:	Adaptive Predictive PID
Degree:	Master of Science
Year this Degree Granted:	1995

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Permanent address:
35 Wedgewood Avenue
Spruce Grove, Alberta
T7X 1M2

Date: March 30, 1995

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Adaptive Predictive PID** submitted by Randy Marvin Miller in partial fulfillment of the requirements for the degree of Master of Science in Process Control.

Dr. S.L. Shah
(co-supervisor)

Dr. R.K. Wood
(co-supervisor)

Dr. D.G. Fisher
(Chemical Engineering)

Dr. Q.-H.M. Meng
(Electrical Engineering)

Date: March 27, 1995

To
my life partner, Laurie
and
my Parents

Abstract

This thesis is primarily concerned with the development and evaluation of an adaptive predictive proportional-integral-derivative (PID) control scheme for control of processes with time varying dynamics and a time varying delay. One of the main motivations for this new PID scheme is to utilize existing industrial control computers that employ a PID algorithm for the implementation of advanced control. Predictive PID constants and the internal model are chosen by equating the discrete PID control law with the linear form of generalized predictive control with steady state weighting. A recursive least squares algorithm based on an upper diagonal factorization method is employed to recursively update the model upon which the predictive PID controller is based. In addition, a novel on-line time delay estimation technique is proposed by rationalization of the coefficients of an extended numerator model. Excellent performance of predictive and adaptive predictive PID with on-line time delay estimation is demonstrated for control of simulated, experimental and industrial processes with time varying dynamics and time delays.

Acknowledgments

This work is the result of considerable guidance and support from many people over the last several years. I am very grateful to Dr. Reg Wood for persuading me to study Chemical Engineering and for his personal mentorship while I was an undergraduate and a graduate student. I am also very grateful to Dr. Sirish Shah for his technical advice and his personal guidance during this work. Special thanks are due to Dr. Grant Fisher for his advice regarding AUDI and critical review of this thesis, to Dr. Ezra Kwok for his assistance in the industrial application of this work and to Walter Boddez for ensuring the flawless operation of the experimental equipment.

I am thankful for the many lively discussions regarding my thesis topic among other issues with my colleges Biao Huang, Pranob Banerjee, Ravindra Gudi, S. (Laksh) Lakshminarayanan, Munawar Saudagar, Mary Bourke and Gary Kwong. Life as a graduate student was made much more enjoyable by my fellow graduate students Kevin Dorma, Birgitte Willumsen, Yves Lacombe, Harvey Yarranton, Alison Miller, Susan Richardson, Doris Leung and Sam Erlenbach.

Financial support from the Department of Chemical Engineering and the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

Above all I would like to express my sincerest gratitude to my wife Laurie who gave me her unconditional support and made countless sacrifices while I was absorbed with this work and asked for nothing in return. My sincerest gratitude also goes out to my parents who always gave me their unconditional love and support throughout my life regardless what endeavor I was involved in.

Contents

1 Introduction 1

- 1.1 History and Background 1
- 1.2 Motivation for the Current Work 3
- 1.3 Scope and Objectives of Thesis 3
- 1.4 Organization of Thesis 4

2 Development of a Stochastic Predictive PID Controller 8

- 2.1 Introduction 9
- 2.2 Generalized Predictive Control (GPC) 10
- 2.3 GPC with Steady State Weighting (γGPC) 13
- 2.4 The Linear Form of GPC 17
- 2.5 The Discrete PID Control Law 19
- 2.6 The PI/PID form of γGPC 20
- 2.7 The Multistep Long Range Predictor, G_{MP} 23
  - 2.7.1 Time Delay Compensation 23
  - 2.7.2 Extension of G_{MP} to Stochastic Disturbance Compensation 26
- 2.8 Interpretations of the Predictor, G_{MP} 32
- 2.9 Infinite Horizon GPC and PID 34
- 2.10 Experimental Evaluation of Predictive PID 40
  - 2.10.1 The Light Bulb Process 40
  - 2.10.2 Open Loop Analysis 41
  - 2.10.3 Closed Loop Comparison of GPC and Predictive PID 44

2.11 Industrial Application of Predictive PID 46
    2.11.1 Process Description 47
    2.11.2 The NO_2 Process Problem 48
    2.11.3 Control Strategy and Implementation 49
    2.11.4 Comparison of PID and Predictive PID 49
2.12 Conclusions 50
References 52

3 Control Analysis with Performance Measures 55
3.1 Introduction 55
3.2 Robustness Measure 56
    3.2.1 The Small Gain Theorem 56
    3.2.2 Robustness Margin 58
3.3 Overall Performance Measure 59
3.4 Examples 60
    3.4.1 Example 1 60
    3.4.2 Example 2 62
    3.4.3 Example 3 63
3.5 Conclusions 64
References 65

4 Adaptive Predictive PID 66
4.1 Introduction 67
4.2 The Predictive PID Control Law 68
4.3 Recursive Plant Model Estimation Using the Augmented Upper Diagonal Identification (AUDI) Method 70
4.4 Adaptive Control Strategy 73
4.5 Experimental Evaluation of Adaptive Predictive PID 74
    4.5.1 The Stirred Tank Heater Experimental Setup 75
    4.5.2 Control Scheme Implementation 76
        4.5.2.1 Selection of Sampling Rate 76
    4.5.3 Open Loop Analysis 77
    4.5.4 Closed Loop Adaptive Predictive PID 80
        4.5.4.1 Hot Water Flow Rate Disturbance 81

4.5.4.2 Level Disturbance 83

4.5.4.3 Discharge Flow Rate Disturbance 84

4.6 Conclusions 86

References 88

5 Recursive Time Delay Estimation **90**

5.1 Introduction 90

5.2 Review of Existing Recursive Time Delay Estimation Techniques 91

5.2.1 Variable Delay Model Structures 92

5.2.2 Distinct Time Delay Estimation 93

5.2.2.1 Variable Regression Estimation of Process Time Delays 94

5.3 Extended Numerator Rationalization for On-line Estimation of Time Delay 96

5.3.1 Development of ENR 96

5.3.2 Statics Interpretation of ENR 102

5.3.3 Practical Implementation of ENR 102

5.3.4 ENR Time Delay Uncertainty Estimate 103

5.4 Simulation Study 106

5.4.1 Persistent Servo Excitation 106

5.4.2 Regulatory Control 110

5.5 Experimental Evaluation of ENR 112

5.5.1 Experimental Equipment 113

5.3.2 Implementation Details and Open Loop Modelling 114

5.5.3 Experimental Results 115

5.6 Conclusions 123

References 124

6 LabVIEW® for Experimental Process Control **126**

6.1 Introduction 126

6.2 General LabVIEW Description and Comments 127

6.2.1 General Description 127

6.2.2 LabVIEW Programming Paradigms 127

6.2.3 Simple Examples 128

6.3 Adaptive LRPC Implementation Issues 129

6.3.1 Program Structure 129

6.3.2 Diophantine Identity 130
6.3.3 GPC Control Law 131
6.3.4 Augmented *UD* Identification (AUDI) Algorithm 131
6.3.5 Data Logging 131
6.3.6 Graphical User Interface 131
6.3.7 Data Acquisition 132
6.3.8 Program Execution and Debugging 133
6.4 Recommendations for Future Development 133
6.5 Conclusions 134
References 135

7 Industrial Batch Identification Study **136**

7.1 Introduction 136
7.2 Batch Least Squares Identification Techniques 137
7.2.1 Process Models 137
7.2.2 Batch Least Squares 138
7.2.3 Augmented Upper Diagonal Identification 139
7.2.4 The Matlab System Identification Toolbox 141
7.3 Identification Case Study 142
7.3.1 ARX Model Estimation 142
7.3.1.1 Selection of Time Delay and Model Order 142
7.3.1.2 Time Domain Validation 147
7.3.2 ARIX Model Estimation 148
7.3.3 ARMAX Model Estimation 148
7.3.3.1 Selection of Model Order and Time Delay 149
7.3.3.2 Time Domain Model Validation 150
7.3.4 Analysis of Residuals 152
7.3.5 Frequency Domain Model Validation 152
7.4 Conclusions 154
References 155

8 Conclusions and Recommendations **156**

8.1 Conclusions 156
8.2 Recommendations 160

List of Figures

2.1 Block diagram of the predictive deterministic PID control loop ... 22

2.2 GPC and equivalent PID control response for a discrete second order plant ... 22

2.3 Block diagram of the predictive model based PID control loop ... 25

2.4 GPC and equivalent PID response to a second order plant ... 26

2.5 Block diagram of the stochastic predictive PID control loop ... 28

2.6 Alternate block diagram of the stochastic predictive PID control loop ... 29

2.7 Response of deterministic GPC and predictive PID to a stochastic process ... 30

2.8 Response of stochastic GPC and predictive PID to a stochastic process ... 30

2.9 Response of GPC and predictive PID with an over estimated C polynomial to a stochastic process ... 31

2.10 Response of GPC and predictive PID with a PI on SP form and an over estimated C polynomial to a stochastic process ... 31

2.11 Response of GPC and predictive PID with a PI on SP form to a stochastic process ... 32

2.12 Block diagram of deterministic predictive PID control loop with the expanded predictor, G_{MP} ... 33

2.13 Block diagram of the stochastic predictive PID control loop with the expanded predictor, G_{MP} ... 34

2.14 response of ∞GPC and ∞PID to a stochastic plant ... 38

2.15 Response of ∞GPC and ∞PID with $C=1+0.4q^{-1}$ 39

2.16 Schematic diagram of the light bulb process 41

2.17 Steady state temperature versus controller output for the light bulb process 42

2.18 Open loop response of the light bulb process 43

2.19 Temperature response to the fan disturbance for the light bulb process 43

2.20 GPC servo and regulatory response to the light bulb process 45

2.21 Predictive PID servo and regulatory response to the light bulb process 45

2.22 ∞PID servo and regulatory response to the light bulb process 46

2.23 Simplified schematic and instrument diagram for the NO_2 process 47

2.24 Comparison of PID and predictive PID for control of the NO_2 process 50

2.25 Extended regulatory comparison of PID and predictive PID for control of the NO_2 process 51

3.1 Feedback control of a modelled plant with additive perturbation 57

3.2 Perturbed feedback loop 57

3.3 Graphical interpretation of the robustness margin 59

3.4 Performance surface for GPC control of a third order plant with $N_u = 1$ 61

3.5 Performance surface for GPC control of a third order plant with $N_u = 2$ 61

3.6 Performance surface for GPC control of a first order process as C_c and the time delay varies 62

3.7 Performance surface for GPC control of a first order process as λ and the time delay varies 63

3.8 Performance surface for GPC control of a first order process with a varying time delay and gain 64

4.1 Block diagram representation of the predictive PID controller 69

4.2 Block diagram of adaptive predictive PID in closed loop 74

4.3 Schematic diagram of the stirred tank heater 75

4.4 Open loop discharge temperature response for the nominal case 78

4.5 Open loop temperature response for the hot water flow rate disturbance 79

4.6 Open loop temperature response for the level disturbances 79

4.7 Open loop temperature response for the discharge flow disturbance 80

4.8 PID control of the stirred tank heater with a hot water flow disturbance 81

4.9 Predictive PID control of the stirred tank heater with a hot water flow disturbance 82

4.10 Adaptive predictive PID control of the stirred tank heater with a hot water flow disturbance 83

4.11 Predictive PID control of the stirred tank heater with a level disturbance 84

4.12 Adaptive predictive PID control of the stirred tank heater with a level disturbance 85

4.13 Predictive PID control of the stirred tank heater with a discharge flow rate disturbance 86

5.1 Recursion of extended numerator models to a time delay varying plant with open loop input excitation 97

5.2 Graphical approximation of fractional time delay by comparing the relative weights of B_d 100

5.3 Application of ENR to a time delay varying first order plant subject to open loop excitation 101

5.4 Statics analogy of ENR 102

5.5 Block diagram representation of ENR in combination with plant model adaptation for an on-line implementation 103

5.6 Adaptive predictive PID in combination with ENR subject to persistent servo excitation 107

5.7 Effect of mismatch in *a* for closed loop on-line time delay estimation using ENR 108

5.8 Adaptive predictive PID in combination with VRE subject to persistent servo excitation 109

5.9 Effect of mismatch in *a* for closed loop on-line time delay estimation using VRE 109

5.10 Adaptive predictive PID in combination with ENR for regulatory control of a noisy process subject to frequent disturbances 111

5.11 Adaptive predictive PID in combination with VRE for regulatory control of a noisy process subject to frequent disturbances 112

5.12 Schematic diagram of the stirred tank heater 113

5.13 Fixed gain predictive PID control performance for a discharge flow rate disturbance 115

5.14 Adaptive predictive PID control performance for a discharge flow rate disturbance 116

5.15 Adaptive predictive PID with ENR control performance for a discharge flow rate disturbance 117

5.16 Adaptive predictive PID control performance for a discharge flow rate disturbance followed by an inlet temperature disturbance 118

5.17 Adaptive predictive PID control performance for a discharge flow rate disturbance followed by an inlet temperature disturbance 119

5.18 Predictive PID control performance for a shift in the plant time delay and a perturbation in the inlet temperature 120

5.19 Adaptive predictive PID with ENR control performance for a shift in the plant time delay and a perturbation in the inlet temperature 121

5.20 Adaptive predictive PID with ENR control performance for a shift in the plant time delay without an inlet temperature perturbation 122

6.1 LabVIEW program to interactively add two numbers 128

6.2 LabVIEW implementation of a digital first order filter 129

6.3 Block diagram of adaptive predictive PID for control of a steam heated stirred tank heater 130

6.4 Front panel of adaptive predictive PID for control of a steam heated stirred tank heater 132

7.1 General process representation 138

7.2 Open loop industrial input-output data 143

7.3 ARX model loss function plots with respect to time delay using the ID toolbox 144

7.4 Impulse response estimate of the ID data set using the ID toolbox function *cra* 145

7.5 Loss function plot with respect to model order for ARX models 146

7.6 Simulated response of the ARIX model 148

7.7 Loss function with respect to model order for the ARMAX model with a first order *C* polynomial 149

7.8 Simulated response of ARX and ARMAX models to all data sets 151

7.9 Unit step responses of plant models 152

7.10 Auto-correlation of residuals for the ID data set 153

7.11 Frequency response of the plant data and models 154

List of Tables

4.1 The Recursive AUDI Algorithm with Variable Forgetting 72

4.2 Nominal operating conditions 76

4.3 Operating conditions for the open loop stirred tank heater runs 77

4.4 Open loop modeling of the stirred tank heater 78

4.5 Experimental closed loop control of the stirred tank heater 81

5.1 The recursive VRE algorithm with data forgetting 95

5.2 Implementation of ENR Based on Recursive AUDI 104

5.3 Nominal operating conditions for the stirred tank heater 114

5.4 Open loop modelling of the stirred tank heater 114

7.1 Optimal ARX model order and time delay using the ID toolbox function *selstruc* 146

7.2 Estimated ARX model parameters using AUDI and the ID toolbox 147

7.3 Time domain validation of ARX model estimates 147

7.4 Optimal ARMAX model order and time delay using the ID toolbox and (7.3.1) 150

7.5 Estimated ARMAX model parameters using the ID toolbox 150

7.6 Time domain validation of the ARMAX model 150

List of Symbols

$A(q^{-1})$	Plant model denominator polynomial
$\bar{a}_1$	Nominal first order coefficient of $A(q^{-1})$
a_i	Element of $A(q^{-1})$
a_m	Denominator coefficient in VRE estimation model
$A_p(q^{-1})$	Actual plant denominator polynomial
$B(q^{-1})$	Plant model numerator polynomial with ZOH removed
$B_d(q^{-1})$	Extended numerator polynomial
b_i	Element of $B(q^{-1})$
$b_{i,CI}$	Confidence interval of the b_i estimate
b_m	Numerator coefficient in VRE estimation model
$B_p(q^{-1})$	Actual plant numerator polynomial with ZOH removed
$C(q^{-1})$	Noise model polynomial
$C(t)$	Covariance matrix
$C_c(q^{-1})$	C polynomial when used as a tuning constant
$C_e(q^{-1})$	Estimation filter
c	Element of $C(q^{-1})$; penalty weight of the number of parameters in the AIC
$C_p(q^{-1})$	Actual noise polynomial
d	Time delay in sample intervals
$\mathbf{d}$	Time delay vector
$D(t)$	Diagonal matrix used in AUDI
$\hat{d}(t)$	Time delay estimate

$d^f(t)$	ENR filtered delay estimate
dim	Max(nA, $nB+d+1$)-1
$dist$	Continuous disturbance variable
$d_{\max}$	Maximum time delay in sample intervals
$d_{\min}$	Minimum time delay in sample intervals
$e(t)$	Error between the setpoint and the feedback
E	Expectation operator
E_0	Expected value of autocorrelation terms in VRE objective function
E_1	Expected value of cross-correlation terms in VRE objective function
e_j	Last coefficient of E_{j+1}
E_j	Polynomial from j step ahead Diophantine identity
e_s	Steady state value of e_j
f	Intermediate variable in the AUDI algorithm
$\mathbf{f}$	Unforced response vector
f_j	j^{th} element of unforced response
F_j	Polynomial from j step ahead Diophantine identity
$\mathbf{f_s}$	Steady state unforced response vector
f_s	First row of $\mathbf{f_s}$
F_s	Steady state value of F_j
F_∞	Converged value of F_j (equivalent to F_s)
g	Intermediate variable in the AUDI algorithm
$\mathbf{G}$	Matrix of $\tilde{G}_j$ elements
G_C	General controller representation
G_{Cw}	Predictive PID servo polynomial
G_{Cw}^d	Predictive PID servo polynomial based on a deterministic model
G_{Cy}	Predictive PID regulatory polynomial
G_{Cy}^d	Predictive PID regulatory polynomial based on a deterministic model
G_d	Disturbance model transfer function
g_j	Last coefficient of $\overline{G}_{j+1}$
$\overline{G}_j$	Polynomial obtained from E_jB
$\tilde{G}_j$	Polynomial obtained from E_jB
G_{MP}	Predictive PID internal model

G_{MP}^{d}	Predictive PID internal model based on a deterministic model
G_p	Plant transfer function
$\hat{G}_p$	Plant model
$\tilde{G}_p$	Additive model plant mismatch
g_s	Steady state value of g_j
$\mathbf{G_s}$	Lower triangular matrix containing only g_s elements
$\overline{G}_s$	Steady state value of $\overline{G}_j$
$\overline{G}_\infty$	Converged value of $\overline{G}_j$ (equivalent to $\overline{G}_s$)
g_∞	Converged or steady state step response coefficient
G_{Cw}^{∞}	Infinite horizon predictive PID servo polynomial
G_{Cy}^{∞}	Infinite horizon predictive PID regulatory polynomial
G_{MP}^{∞}	Infinite horizon predictive PID internal model
$\mathbf{h}$	First row of $\left[\mathbf{G}^T\Gamma_y\mathbf{G}+\Lambda+\mathbf{G}_s^T\Gamma\mathbf{G}_s\right]^{-1}\mathbf{G}^T\Gamma_y$
h_s	Sum of the elements of the first row of $\left[\mathbf{G}^T\Gamma_y\mathbf{G}+\Lambda+\mathbf{G}_s^T\Gamma\mathbf{G}_s\right]^{-1}\mathbf{G}_s^T\Gamma$
i^*	ENR solution of extended numerator index such that $\hat{d}(t)=d_{\min}+i^*$
J	Loss function or objective function
J_{mod}	Loss function with a penalty on the number of parameters in the model
K_D	Discrete PID derivative tuning constant
K_{D_c}	Continuous PID derivative tuning constant
K_I	Discrete PID integral tuning constant
K_{I_c}	Continuous PID integral tuning constant
K_P	Discrete PID proportional tuning constant
K_{P_c}	Continuous PID proportional tuning constant
ℓ	Time delay rounded to next highest integer
L	Loss function of AUDI backward model
m	Fractional part of time delay such that $d=\ell - m$
M	General feedback loop transfer function
M_{GPC}	GPC feedback loop transfer function
M_{PID}	PID feedback loop transfer function
M_{PPID}	Predictive PID feedback transfer function
MSF	Mean square fit
N	Asymptotic data length used in computation of the variable forgetting factor

N_1	Minimum prediction horizon
N_2	Maximum prediction horizon
n	Model order
nA	Order of $A(q^{-1})$
nB	Order of $B(q^{-1})$ with plant ZOH removed
nC	Order of $C(q^{-1})$
nd	$d_{\max} - d_{\min} + 1$
N_p	Number of parameters in the plant model
N_u	Control horizon
q^{-1}	Backward shift operator
r_i	Element of $R(q^{-1})$
$R(q^{-1})$	Polynomial in the linear form of GPC or γGPC
RM	Robustness margin
s_i	Element of $S(q^{-1})$
$S(q^{-1})$	Polynomial in the linear form of GPC or γGPC
$T(q^{-1})$	Polynomial in the linear form of GPC or γGPC
$t_{c,N-1}$	Critical value from t distribution tables
$T_p(q^{-1})$	Equivalent to T-1
T_s	Sampling interval
$u(t)$	Controller output or process input
$U(t)$	Upper diagonal matrix used in AUDI
$u^f(t)$	Controller output filtered by $1/C$
$\tilde{\mathbf{u}}$	Present and future incremental control elements
$\mathbf{w}$	Vector of future setpoints
$w(t)$	Setpoint
w_s	Steady state setpoint
$x(t)$	Disturbance term
y	Continous process output
$y(t)$	Discrete process output
$\tilde{\mathbf{y}}$	Vector of predicted plant output values
$\hat{y}(t+j\|t)$	Prediction of $y(t+j)$ conditioned on data up to t
$y^f(t)$	Plant output filtered by $1/C$

$\tilde{\mathbf{y}}_s$	Vector of predicted steady state plant output values
z	Additive model plant mismatch output

Greek Symbols

α	Filter constant in ENR
$\hat{\alpha}(t)$	Backward model from AUDI
Δ	Differencing operator, $1-q^{-1}$
$\varepsilon(t)$	Prediction error
$\phi(t)$	Augmented regressor for use with AUDI
$\phi_{BLS}(t)$	Regressor used in batch least squares
γ	Infinite horizon or steady state weighting term
Γ	Infinite horizon or steady state weighting matrix
γ_y	Finite horizon weighting term
Γ_y	Finite horizon weighting matrix
λ	Control weighting term
$\lambda(t)$	Variable forgetting factor used in AUDI
λ_p	Incremental control penalty in the overall performance measure
Λ	Control weighting matrix
μ	Intermediate variable in the AUDI algorithm
ν	Intermediate variable in the AUDI algorithm
$\hat{\theta}(t)$	Augmented parameter vector used in AUDI
$\hat{\theta}_{BLS}(t)$	Parameter vector used in batch least squares
$\sigma_{b_i}^2$	Variance of b_i coefficient
σ_d^2	Variance of the time delay estimate
σ_ξ^2	Noise variance
τ	Dominant process time constant
ω	Frequency in radians
$\xi(t)$	Uncorrelated random sequence of zero mean
ψ	Robustness margin penalty in the overall performance measure

Abbreviations

AIC	Akaike's information theoretic criterion
ARIMAX	Auto regressive integrating moving average with exogenous input
ARMAX	Auto regressive moving average with exogenous input
ARX	Auto regressive with exogenous input
BLS	Batch least squares
CE	Characteristic equation
cvs	Control valve signal
DMC	Dynamic matrix control
EE	Equation error model
ENR	Extended numerator rationalization
GPC	Generalized predictive control
IDCOM	Identification/command
IAE	Integral of the absolute errors
IDTB	Identification toolbox
ISE	Integral of the square of the error
ITAE	Integral of the time weighted absolute errors
LQG	Linear quadratic Gaussian
LRPC	Long range predictive control
PID	Proportional integral derivative
γGPC	Generalized predictive control with steady state weighting
∞GPC	Infinite horizon generalized predictive control
MVC	Minimum variance control
PID	Infinite horizon proportional integral derivative
SGT	Small gain theorem
TF	Transfer function
VRE	Variable regression estimation

Chapter 1

Introduction

The focus of this thesis is on the development of a model based adaptive PID controller suitable for control of processes with time varying dynamics and time varying delays. This chapter gives a brief background into PID control and provides the motivation for the work presented in subsequent chapters.

1.1 History and Background

Prior to 1940, most processes in industrial chemical plants were manually operated while simple proportional feedback control was used only in a select few applications. Large surge vessels located between process units and numerous human operators were required for the continuous operation of even simple chemical plants. During the 1940s and 1950s, the development of more efficient higher capacity chemical process equipment and higher labour costs made manual control either uneconomical or impossible (Luyben, 1990). Subsequently, the proportional-integral-derivative (PID) feedback controller was used in numerous industrial applications which motivated the development of rules or techniques for its tuning (Ziegler and Nichols, 1942; Cohen and Coon, 1953; Lopez *et al.*, 1967; Rivera *et al.*, 1986). Significant time delays in the process proved to be a major problem for a PID controller which prompted the development of time delay compensation techniques such as the Smith predictor (Smith, 1957). Even during the 1970s when energy costs rose dramatically and inexpensive digital computers were produced, classical PID remained the controller of choice in industry. A new class of adaptive predictive controllers based on *d* step ahead predictions (where *d* is the time delay) evolved in the 1970s starting with the developments of Åström and Wittenmark (1973) and Clarke and Gawthrop (1975). Subsequently, controllers based on long range multistep predictions (Richalet et al., 1987; Cutler and Ramaker, 1980) ensued. This lead to the generalized predictive controller (GPC) (Clarke *et al.*, 1987) which unified previous long range predictive control (LRPC) strategies. Numerous reported applications (Shah, 1994) and many more unreported applications indicate the unprecedented success of LRPC in industry, particularly for difficult control problems. However, one of the drawbacks of LRPC is that specialized software is required and the computational demands often require additional expensive computer hardware. Although LRPC and other advanced controllers have managed to capture a portion of the industrial control applications, PID is still by far the most popular control algorithm used today (Åström and Hägglund, 1988; Fisher 1991). Instrument and control engineers alike are familiar with the entrenched philosophy of PID control. Consequently, control computers that employ a conventional PID algorithm are manufactured in the hundreds of thousands every year (Åström and Hägglund, 1988) which suggests the continued popularity of the conventional PID controller for the foreseeable future.

Adaptive LRPC strategies have typically focused on the recursive estimation of a plant model with a fixed time delay in conjunction with the execution of an LRPC algorithm based on the plant model. For processes with time varying dynamics or nonlinear processes, adaptive LRPC can potentially provide improved closed loop performance compared to a fixed model controller. However, it was shown by Shook et al. (1992) that the identification strategy must be compatible with the control strategy to meet the objectives of the adaptive controller. Furthermore, Seborg *et al.* (1989) suggests that *ad-hoc* developments have had a significant impact on the success of adaptive controllers in industry. Practical issues regarding the industrial implementation of adaptive advanced control remains a relatively wide open area.

Many processes are known to exhibit a significantly varying time delay due to changing flow rates or mixing conditions (Dumont *et al.*, 1993). Since the time delay element contributes significantly to the phase of the closed loop, it has a major effect on the closed loop stability. For PID control applications it is always safe to base the controller constants on an overestimated time delay which results in detuned but stable performance. However, for LRPC and other advanced controllers, the time delay cannot be simply overestimated because this may result in unstable or unacceptable performance. It is therefore advantageous that the time delay and the process dynamics are simultaneously estimated on-line in a continuous or on-demand basis for adaptive control of time variant processes.

1.2 Motivation for the Current Work

The two primary motives of this thesis are:

1.) To facilitate the implementation of advanced control in existing industrial control computers using a PID structure.

2.) Development of a practical identification strategy for adaptive control of processes with varying dynamics and time delays.

1.3 Scope and Objectives of Thesis

This thesis is primarily concerned with the development and analysis of a model based adaptive PID controller. Because GPC with steady state weighting (Kwok and Shah, 1994) is a useful enhancement of the traditional finite-horizon LRPC, it was used as the basis for a new model based or "predictive" PID controller. Since the augmented upper diagonal factorization technique (denoted as AUDI; Niu *et al.*, 1992) is an elegant and efficient reformulation of recursive least squares identification, it was chosen for the parameter adaptation of adaptive predictive PID. Although numerous time delay estimation techniques are reported in the literature (Ferretti *et al.*, 1991), a new method is proposed here to take advantage of the practical features within the AUDI framework (Niu *et al.*, 1994; Niu *et al.*, 1995). Thus, the final objective of this work is the development and evaluation of an adaptive model based PID controller capable of controlling processes with time varying dynamics as well as a time varying delay.

The main contributions of this thesis are:

- development of a model based predictive PID controller that is mathematically equivalent to GPC and the variants of GPC for low order processes,

- implementation of a predictive PID control algorithm in an industrial control computer that employs a conventional PID algorithm and evaluation of the performance of a predictive PID controller for control of an industrial process,

- development of an overall performance measure which combines traditional measures of performance with a measure of robustness for assessment of control performance,

- development of a recursive time delay estimation technique based on the rationalization of a time series model with a high order numerator,
- experimental evaluation of adaptive predictive PID with time delay estimation for control of a process with varying dynamics and a varying time delay,
- batch identification case study using industrial open loop data.

1.4 Organization of Thesis

Since many of the chapters of this thesis have been submitted for publication or are planned for future publication, each chapter is self contained with its own introduction and conclusion section. The chapters are organized as follows.

Chapter 2 describes the formulation of a predictive PID controller from the linear form of the GPC control law with steady state weighting. The resulting internal model of predictive PID is interpreted as a multistep weighted predictor that is optimal in terms of the GPC objective function. Extensions of predictive PID are described by selectively choosing the GPC controller constants. The equivalence between GPC and predictive PID is demonstrated through simulations and an experimental application. An industrial application is also presented to show the practical effectiveness of a predictive PID controller compared to a conventional PID controller.

Chapter 3 presents an overall performance measure for assessment of control behavior that consists of an absolute performance measure subject to a penalty on the incremental control action and the inverse of a new margin of robustness. The small gain theorem is used to determine the new scalar robustness margin for a feedback control loop with a known model plant mismatch.

Chapter 4 extends the predictive PID controller to an adaptive predictive PID controller by combining the results of chapter 2 with the AUDI algorithm. It is shown that bandpass filtering of the input-output data is required so that the long range objectives of predictive PID are compatible with the one step ahead objective of AUDI. An experimental evaluation of adaptive predictive PID for control of a process with linear but time varying dynamics is conducted.

Chapter 5 introduces a new method of on-line time delay estimation based on the rationalization of an extended numerator (denoted as ENR) model which is estimated independently from the plant model by a second AUDI algorithm. A method of moments is used to extract a distinct time delay in terms of sampling intervals from the numerator coefficients of the model. Uncertainty of the time delay estimate is computed from the propagation of variance of the numerator coefficients. Experimental results of adaptive predictive PID with ENR for control of a process that exhibits significantly varying dynamics and a time varying delay are presented.

Chapter 6 discusses the graphical development and data acquisition software, LabVIEW® (Anon, 1993), for the experimental implementation of process control. Experience from the experimental implementation in this work as well as teaching assistant experience in an undergraduate real time computing course primarily motivate the discussion in this chapter.

Chapter 7 compares the batch AUDI algorithm with the Matlab® System Identification Toolbox (Ljung, 1992) for the identification of a time series model from open loop industrial data. Time and frequency domain techniques are used to validate the models on two different open loop data sets compared to the identification data set.

Chapter 8 summarizes the results of this thesis and suggests several areas for future work.

References

Anon, *LabVIEW for Windows User Manual*, National Instruments Corporation, Austin, TX, 1993.

Åström, K.J., and B. Wittenmark, "On self-tuning regulators," Automatica, Vol. 9, No. 2, pp 185-199, 1973.

Åström, K.J., and T. Hägglund, *Automatic Tuning of PID Controllers*, ISA, NC, 1988.

Cohen, G.H., and G.A. Coon, "Theoretical consideration of retarded control," *Trans. ASME*, **75**, pp 827-834, 1953.

Clarke, D.W., and P.J. Gawthrop, "Self-tuning controller," *IEE Proc. D*, **122** (9), pp 929-934, 1975.

Cutler, C.R., and B.L. Ramaker, "Dynamic matrix control -- a computer control algorithm," *Proc. American Control Conf.*, San Francisco, CA, 1980.

Dumont, G.A., A. Elnaggar and A. Elshafei, "Adaptive predictive control of systems with time-varying time delay," *Int. J. Adaptive Control and Signal Processing*, Vol. 7, pp 91-101, 1993.

Ferretti, G., C. Maffezzoni and R. Scattolini, "Recursive estimation of time delay in sampled systems," *Automatica*, Vol. 27, No. 4, pp 653-661, 1991.

Fisher, D.G., "Process control: an overview and personal perspective," *Can. J. Chem. Eng.*, **69,** pp 5-26, 1991.

Kwok, K., and S.L. Shah, "Long-range predictive control with a terminal matching condition," *Chemical Engineering Science*, vol **49**, No. 9, pp 1287-1300, 1994.

Ljung, L., *System Identification Toolbox for use with Matlab*, The MathWorks, Inc., Natick, Mass., 1992.

Lopez, A.M., P.W. Murrill and C.L. Smith, "Controller tuning relationships based on integral performance criteria," *Instrumentation Technology*, **14** (11), pp 57-67, 1967.

Luyben, W.L., *Process Modeling, Simulation, and Control for Chemical Engineers*, McGraw-Hill, New York, 1990.

Niu, S., D.G. Fisher and D. Xiao, "An augmented UD identification algorithm," *Int. J. Control*, Vol. 56, No. 1, pp. 193-211, 1992.

Niu, S. and D.G. Fisher, "Monitoring parameter identifiability during on-line identification," *presented at the 3rd IEEE Conf. Control Applications (also submitted to Int. J. Cont.)*, Glasgow, August, 1994.

Niu, S. and D.G. Fisher, "Simultaneous estimation of process parameters, noise variance and signal-to-noise ratio," *IEEE Trans. Signal Processing*, in press, 1995.

Richalet, J., A. Rault, J.L. Testud, J. Papon "Model Predictive heuristic Control: Applications to Industrial Processes," *Automatica*, Vol. 14, No. 5, pp 413-428, 1978.

Rivera, D.E., M. Morari and S. Skogestad, "Internal model control. 4 PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, **25**, pp 252-265, 1986.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp, *Process Dynamics and Control*, pp 272-309, John Wiley and Sons, New York, 1989.

Shah, S.L., "Model-based predictive control: theory and implementation issues," *presented at ADCHEM '94*, Kyoto Research Park, Kyoto, Japan, 1994.

Shook, D.S., C. Mohtadi and S.L. Shah, "A control relevant identification strategy for GPC," *IEEE Trans. Automatic Control*, Vol. 37, No. 7, pp 975-980, 1992.

Smith, O.J.M., "Closer control of loops with dead time," *Chemical Engineering Progress*, **53** (5), pp 217-219, 1957.

Ziegler, J.G., and N.B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, **64**, pp 759-768,1942.

Chapter 2

Development of a Stochastic Predictive PID Controller[1]

A new stochastic predictive proportional-integral-derivative (PID) control law is proposed which is mathematically equivalent to generalized predictive control (GPC) with a terminal matching condition. The main motivation of this chapter is the extension of the classical PID algorithm on industrial computers to do advanced control without employing specialized software. The predictive PID constants and the internal model are chosen by equating the discrete PID control law with the linear form of GPC. The result is a long range predictive control law with a model based PID structure. Predictive PID is stochastic because GPC is based on an ARIMAX model of the plant plus the noise term. A first order model yields a PI controller while a second order plant results in a PID structure. The process model order is restricted to a maximum of two although there is no restriction on the choice of GPC tuning parameters. Stochastic disturbances are handled through solution of the appropriate Diophantine identities. Performance of predictive PID scheme is shown, via simulation, to be identical to GPC. Results from applying the predictive PID algorithm for the control of a laboratory process and a key industrial heat exchanger are presented.

---

[1] A version of this chapter has been accepted for presentation as: Miller R.M., K.E. Kwok, S.L. Shah and R.K. Wood, "Development of a Stochastic Predictive PID Controller", *Proc. American Control Conference*, Seattle, WA, 1995.

2.1 Introduction

It is well known that most industrial process control applications implement a PID feedback strategy (Åström and Hägglund, 1988; Gawthrop, 1986; Fisher, 1991). The PID controller is well understood and accepted among operations personnel and control engineers due to the intuitive simplicity of the algorithm. Explicit knowledge of the process model is not required to successfully implement PID control on many chemical processes. In addition, the dynamics of many processes are well suited to PID control (Gawthrop, 1986). The *ad-hoc* nature of tuning a PID control loop is an advantage for easy control situations, however, this becomes a disadvantage when faced with a difficult control problem. A PID controller must be significantly detuned to remain stable for adversities such as long time delays or non minimum phase plants which results in poor performance. Mediocre process regulation is no longer acceptable or profitable for many key processes. The recent focus on global markets and global competition for many industries has made current product yields and efficiencies unprofitable in many cases. These business conditions have expanded the requirement for tight regulation beyond the capabilities of simple PID control particularly for key processes. Advanced control schemes such as long range predictive control (LRPC) and linear quadratic Gaussian (LQG) control use plant models in the control law which results in improved control for systems that are otherwise difficult to control using PID. The advanced schemes can also compensate for stochastic disturbances by including the noise model in the control law which results in improved regulation and fewer control moves. A disadvantage of these advanced controllers is the mathematical complexity and the expense of commercial software.

This chapter first extends standard GPC theory to include a steady state weighting term, γ, in the objective function (and hence denoted as γGPC) in order to formulate a basis for predictive PID. Readers familiar with γGPC may wish to advance to section 2.5. The conditions under which the linear form of γGPC and a discrete incremental PID control law are equivalent are established in section 2.5. An internal model for the PID controller is developed which provides multistep long range prediction to compensate for time delays and stochastic disturbances. The polynomials in the linear γGPC control law are used to determine the PID constants and internal model for the predictive PID controller. The result is a model based PID controller that is equivalent to γGPC for regulatory control. This will prove to be a useful result because a γGPC equivalent PID controller can be implemented on existing devices without the expense and complexity of specialized software. Performance of the proposed PID controller is evaluated via simulation for various plant models. Special cases of the model based PID controller are investigated by placing restrictions on the γGPC tuning parameters which form the basis for the PID controller.

2.2 Generalized Predictive Control (GPC)

Although advanced control applications form a small fraction of industrial control systems, LRPC is one of the most popular advanced techniques. The recent use of LRPC for difficult industrial control problems is well noted in the literature (Cutler and Hawkins, 1987; Cutler and Finlayson, 1988; Cutler and Hawkins, 1988; Park, 1988; Hokanson *et al.*, 1989; Tran and Cutler, 1989; Allison *et al.*, 1991; Van Hoof *et al.*, 1989; Shah, 1994). All LRPC methods optimize a quadratic cost function over a multistep or finite-time horizon. The notion of a prediction horizon and a control horizon are integral components of the LRPC objective function. Early LRPC strategies were based on step-response or finite impulse response models as in IDCOM (Richalet *et al.*, 1978) and DMC (Cutler and Ramaker, 1979). Models containing as many as 50 step response coefficients may be required to adequately represent the process dynamics. This fact constrains DMC to nonadaptive control (an adaptive implementation requires recursive updating of up to 50 model parameters). DMC is restricted to stable plants with the exception of integrating plants which are handled through the use of impulse response models. Even if the process is known exactly, the step response model is only an approximation. In addition, the DMC algorithm is not well suited to stochastic disturbance compensation because the noise model is not a component of the DMC control law. In spite of these shortcomings, DMC has been proven to be effective in industry and its documented use is increasing dramatically.

Generalized predictive control is based on an auto regressive integrated moving average with exogenous input (ARIMAX) model of the process. The ARIMAX model can adequately describe most process dynamics using three to five parameters, therefore, GPC is well suited to adaptive control. In addition, unlike DMC, GPC is not restricted to stable plant models because the ARIMAX model can effectively represent stable as well as unstable plants. The moving average (MA) term in the ARIMAX model is the noise model of the plant, hence, stochastic disturbance compensation is an integral part of the GPC control law. The closed loop transfer function of GPC follows naturally from the linear representation of the control law which allows a thorough theoretical analysis of robustness as noted in the literature (Bitmead *et al.*, 1990;

McIntosh *et al.*, 1991 and Banerjee and Shah, 1995). In comparison, DMC has only been subjected to limited theoretical analysis because the characteristic equation is cumbersome (McIntosh *et al.*, 1991).

The GPC law is chosen as a basis for predictive PID for two major reasons. First, GPC has numerous desirable theoretical features as previously mentioned and secondly, GPC is a "generalized" LRPC strategy which includes all desirable properties of DMC. The recent use of DMC in industry suggests that DMC is a proven successor to PID for control problems not well suited to PID.

The derivation of the GPC law introduced by Clarke *et al.* (1987) begins by assuming that the process model can be approximated by the following linear ARIMAX model.

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta}\xi(t) \tag{2.2.1}$$

where A, B, and C are polynomials in the backward shift operator q^{-1} and y, u, and ξ are the predicted output, control input and zero mean white noise disturbance, respectively. In the material that follows, the (q^{-1}) notation is omitted for brevity. The actual process model polynomials, A_p, B_p and C_p are typically of high dimensions, hence, A, B and C are their corresponding low order approximations. The differencing operator Δ (or $1-q^{-1}$) in the plant model ensures that the GPC control law has an integrator for offset elimination. An integrator in the disturbance term also represents realistic disturbances such as random walk or "Brownian motion" type disturbances. The j^{th} step ahead prediction requires the solution of E_j and F_j in the Diophantine identity given by

$$C = E_j A\Delta + q^{-j} F_j \tag{2.2.2}$$

The j^{th} solution of the Diophantine identity (2.2.2) is equivalent to the j^{th} step of the long division of C by $A\Delta$ where E_j and $\frac{F_j}{A\Delta}$ represent the quotient and remainder terms, respectively. Multiplying both sides of (2.2.1) by $E_j\Delta$ results in

$$E_j\Delta A y(t+j) = E_j B\Delta u(t+j-1) + E_j C\xi(t+j) \tag{2.2.3}$$

Substitution of (2.2.2) into (2.2.3) yields the plant model

$$y(t+j)=\frac{E_j B\Delta u(t+j-1)}{C}+\frac{F_j y(t)}{C}+E_j\xi(t+j) \tag{2.2.4}$$

The term $E_j B$ in (2.2.4) can be expressed in a second Diophantine identity as

$$E_j B = C\tilde{G}_j + q^{-j}\overline{G}_j \tag{2.2.5}$$

noting that $\xi(t+j)$ is a zero mean white noise sequence with an expected value of zero. Substitution of (2.2.5) into (2.2.4) and then taking the expected value results in the optimal ARIMAX predictor at time $t+j$ which is expressed as

$$\hat{y}(t+j|t)=\tilde{G}_j\Delta u(t+j-1)+f_j \tag{2.2.6}$$

where

$$f_j=\frac{\overline{G}_j\Delta u(t-1)}{C}+\frac{F_j y(t)}{C} \tag{2.2.7}$$

The notation $\hat{y}(t+j|t)$ reads as follows: the prediction at time t of $y(t+j)$ conditioned on data up to time t. Since (2.2.6) can be separated into a forced term, which depends on present and future Δu, and an unforced term, which depends only on past Δu and y, it is convenient to write the $j=1,\cdots,N$ multistep predictor in the compact vector form as

$$\hat{\mathbf{y}}=\mathbf{G}\tilde{\mathbf{u}}+\mathbf{f} \tag{2.2.8}$$

where

$$\begin{aligned}
\hat{\mathbf{y}}&=\left[\hat{y}(t+1|t),\hat{y}(t+2|t)\quad\cdots\quad\hat{y}(t+N|t)\right]^T\\
\mathbf{G}&=\left[\tilde{G}_1,\tilde{G}_2\quad\cdots\quad\tilde{G}_N\right]^T\\
\tilde{\mathbf{u}}&=\left[\Delta u(t),\Delta u(t+1)\quad\cdots\quad\Delta u(t+N-1)\right]^T\\
\mathbf{f}&=\left[f(t+1),f(t+2)\quad\cdots\quad f(t+N)\right]^T
\end{aligned} \tag{2.2.9}$$

Derivation of the GPC control law follows from the minimization of a receding horizon cost criterion composed of a prediction error squared term and an incremental control squared term.

$$J=E\left\{\sum_{j=N_1}^{N_2}\left[\hat{y}(t+j|t)-w(t+j)\right]^2+\sum_{j=1}^{N_u}\lambda(j)\left[\Delta u(t+j-1)\right]^2\right\} \tag{2.2.10}$$

where $w(t+j)$ is the setpoint, $\lambda(j)$ is the control weight, N_1 is the minimum output prediction horizon, N_2 is the maximum output prediction horizon and N_u is the control horizon. Substitution of (2.2.8) into (2.2.10) and assuming a constant control weighting sequence allows the cost function to be written in a compact form as

$$J = \left\{ (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} \right\} \tag{2.2.11}$$

To find the minimum J with respect to the future control trajectory, Δu one needs to obtain $\frac{\partial J}{\partial \Delta u}$. This results in the GPC control law

$$\tilde{\mathbf{u}} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda \mathbf{I}\right)^{-1} \mathbf{G}^{\mathrm{T}} (\mathbf{w} - \mathbf{f}) \tag{2.2.12}$$

In the implementation of GPC, solving the Diophantine identities (2.2.2) and (2.2.5) are the most computationally intensive steps. An obvious solution to (2.2.2) and (2.2.5) is to perform a deconvolution of $\frac{C}{A\Delta}$ and $\frac{E_j B}{C}$ for N_2 steps. For non adaptive control, (2.2.2) and (2.2.5) need to be solved only once. However, for adaptive control, a solution for every model update is required. Several methods have been proposed to reduce the computational load of GPC in an adaptive implementation. A recursive implementation of the Diophantine identity is investigated in Clarke *et al.* (1987a) and McIntosh (1988). An efficient method of evaluating the unforced response, **f**, without solving the Diophantine identity is proposed by Mutha (1990). A reduced Diophantine formulation by Saudagar *et al.* (1994) recognizes the redundancy of solving $\frac{C}{A\Delta}$ followed by $\frac{E_j B}{C}$ and lastly filtering $y(t)$ and $\Delta u(t-1)$ by $\frac{1}{C}$. Only one Diophantine identity is required in this reduced formulation which reduces the computational load significantly.

2.3 GPC with Steady State Weighting (γGPC)

In this section standard GPC theory is extended to include a steady state or γ weighting term based on the development of Kwok (1992) (also see Kwok and Shah, 1994). A terminal matching condition is included in the GPC cost function which is defined as the weighted square of the steady state error. Thus, GPC with γ weighting is obviously restricted to stable plants because unstable plants do not have a steady state. The notation γGPC will be used to indicate GPC with a steady state weighting term. The steady state prediction is an essential result and this section delves into some detail regarding its derivation.

The j step ahead ARIMAX predictor can be written as

$$\hat{y}(t+j|t) = \tilde{G}_j \Delta u(t+j-1) + \overline{G}_j \Delta u^f(t-1) + F_j y^f(t) \tag{2.3.1}$$

where Δu^f and y^f denote that Δu and y are filtered by $\frac{1}{C}$. Perhaps the most obvious method of obtaining the steady state prediction would be to iterate the Diophantine identities (2.2.2) and (2.2.5) until $\hat{y}(t+j|t)$ converges to $\hat{y}(t+\infty|t)$. However, this would be a computationally expensive and impractical operation. Kwok and Shah (1994) suggest another method to compute the steady state prediction without iteration. Since the polynomials F_j and $\overline{G}_j$ are defined by the identities (2.2.2) and (2.2.5), respectively the $j+1$ polynomials can be written as

$$C = E_{j+1} A\Delta + q^{-j-1} F_{j+1}\,. \tag{2.3.2}$$

and

$$E_{j+1} B = C\tilde{G}_{j+1} + q^{-j-1}\overline{G}_{j+1} \tag{2.3.3}$$

Subtracting equations (2.2.2) and (2.2.5) from (2.3.2) and (2.3.3), respectively gives

$$e_j A\Delta = F_j - q^{-1} F_{j+1} \tag{2.3.4}$$

and

$$e_j B = g_j C + q^{-1}\overline{G}_{j+1} - \overline{G}_j \tag{2.3.5}$$

where e_j is the last coefficient in E_{j+1} and g_j is the last coefficient in $\overline{G}_{j+1}$. As the j^{th} iteration approaches the steady state, (2.3.4) and (2.3.5) converge to

$$e_j A\Delta = e_s A\Delta = F_s - q^{-1} F_s \tag{2.3.6}$$

and

$$e_j B = e_s B = g_s C + q^{-1}\overline{G}_s - \overline{G}_s \tag{2.3.7}$$

Solving for e_s and g_s results in

$$e_s = \frac{C(1)}{A(1)} \tag{2.3.8}$$

$$g_s = \frac{B(1)}{A(1)} \equiv \text{s.s process gain} \tag{2.3.9}$$

Solving for F_s and $\overline{G}_s$ yields

$$F_s = e_s A \tag{2.3.10}$$

$$\overline{G}_s \Delta = g_s C - e_s B \tag{2.3.11}$$

$\overline{G}_s$ can be solved by comparing the coefficients of the left and right terms of (2.3.11).

$$\overline{G}_{s,i} = e_s \sum_{j=i+1}^{nB} b_j - g_s \sum_{j=i+1}^{nC} c_j \tag{2.3.12}$$

which has the equivalent form (because of cross terms)

$$\overline{G}_{s,i} = g_s \sum_{j=0}^{i} c_j - e_s \sum_{j=0}^{i} b_j \tag{2.3.13}$$

From (2.3.10) and (2.3.11), the orders of F_s and $\overline{G}_s$ are nA and max(nB-1, nC-1), respectively. The orders of the ARIMAX model polynomials A, B and C are nA, nB and nC, respectively. With the use of (2.3.10) and (2.3.11), the optimal steady state predictor is found by taking the limit of (2.3.1) as j approaches infinity, that is

$$\lim_{j \to \infty} \hat{y}(t+j|t) = \hat{y}(s|t) = g_s \sum_{j=1}^{Nu} \Delta u(t+j-1) + \overline{G}_s \Delta u^f(t-1) + F_s y^f(t) \tag{2.3.14}$$

Summation in (2.3.14) is required for book keeping purposes when $N_u > 1$ (see the remarks at the end of this section). The control objective function proposed by Kwok and Shah (1994) contains the first two terms of the GPC cost function plus a steady state prediction error term which is based on the optimal steady state predictor (2.3.14). The cost function to be minimized can be expressed as

$$J = \sum_{j=N_1}^{N_2} \gamma_y(j)\left[\hat{y}(t+j) - w(t+j)\right]^2 + \sum_{j=1}^{Nu} \lambda(j)\left[\Delta u(t+j-1)\right]^2 + \sum_{j=1}^{Nu} \gamma(j)\left[\hat{y}(s|t+j-1) - w(s)\right]^2 \tag{2.3.15}$$

where $\gamma_y(j)$ is the output weighting sequence and $\gamma(j)$ is the steady state error weighting sequence. Summation of the steady state term is required because the steady state prediction requires the sum of the first N_u consecutive control responses. Rewriting (2.3.15) in compact form with constant weighting terms yields the following objective function.

$$J = [\hat{\mathbf{y}} - \mathbf{w}]^T \Gamma_y [\hat{\mathbf{y}} - \mathbf{w}] + \tilde{\mathbf{u}}^T \Lambda \tilde{\mathbf{u}} + [\hat{\mathbf{y}}_s - \mathbf{w}_s]^T \Gamma [\hat{\mathbf{y}}_s - \mathbf{w}_s] \tag{2.3.16}$$

where $\hat{\mathbf{y}}$ and $\tilde{\mathbf{u}}$ are defined by (2.2.8) and

$$\begin{aligned} \Gamma_y &= \gamma_y I_{N_2-N_1+1\times N_2-N_1+1} \\ \Lambda &= \lambda I_{N_u\times N_u} \\ \Gamma &= \gamma I_{N_u\times N_u} \end{aligned}$$

and

$$\hat{\mathbf{y}}_s = \mathbf{G}_s \tilde{\mathbf{u}} + \mathbf{f}_s \tag{2.3.17}$$

where

$$\hat{\mathbf{y}}_s = \left[\hat{y}(s|t)\ \hat{y}(s|t+1) \cdots \hat{y}(s|t+Nu-1)\right]^T{}_{N_u\times 1}$$

$$\mathbf{G}_s = \begin{bmatrix} g_s & 0 & \cdots & 0 \\ g_s & g_s & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ g_s & \cdots & \cdots & g_s \end{bmatrix}_{N_u\times N_u}$$

$$\mathbf{f}_s = \left\{[1\ 1 \cdots 1]^T \left[\overline{G}_s \Delta u^f(t-1) + F_s y^f(t)\right]\right\}_{N_u\times 1}$$

Minimization of (2.3.16) yields the main result of this section — the γGPC control law.

$$\tilde{\mathbf{u}} = \left[\mathbf{G}^T \Gamma_y \mathbf{G} + \Lambda + \mathbf{G}_s^T \Gamma \mathbf{G}_s\right]^{-1} \left[\mathbf{G}^T \Gamma_y (\mathbf{w} - \mathbf{f}) + \mathbf{G}_s^T \Gamma (\mathbf{w}_s - \mathbf{f}_s)\right] \tag{2.3.18}$$

Remarks:

- Increasing γ in the γGPC control law has the same control effect as increasing N_2 in the GPC control law when N_u = 1. Therefore, using γ weighting and a small prediction horizon, N_2 gives the same result as a larger value of N_2 without a γ weight. This has the benefit of reducing the computational load by reducing the number of deconvolutions required in the Diophantine identities. Adaptive control is an obvious application of γGPC.

- Several tuning strategies for γ weighting in GPC are presented in Kwok, 1992 and Kwok and Shah, 1994. As a tuning parameter, γ has the benefit of not requiring scaling. In comparison, the effect of control weighting λ in GPC is a function of the process model and therefore requires scaling.
- γ weighting has a stabilizing effect on closed loop control. Several stability properties are discussed in Kwok and Shah, 1994.
- It is obvious from (2.3.18) that GPC is a subset of γGPC since if $\gamma = 0$, γGPC (2.3.18) reduces to GPC (2.2.12).
- An infinite prediction horizon (only when $N_u = 1$) is accomplished by setting λ and γ_y to zero and γ to some positive non-zero value. This is equivalent to setting N_2 to infinity. Both schemes are equivalent to mean level control. Section 2.8 investigates the special case of setting γ_y to zero in more detail.
- Although the γGPC control law accommodates values of N_u over one for non zero values of γ, it is not recommended. Increasing γ when $N_u > 1$ on the γGPC controller does not have the same control effect as increasing N_2 on the GPC controller.

2.4 The Linear Form of GPC

The γGPC control law is implemented in a receding horizon. At each interval, the control vector, $\tilde{\mathbf{u}}$, is computed for the control horizon $t, t+1, \cdots t+N_u-1$ based on the prediction horizon $t+N_1, \cdots t+N_2$. For the receding horizon implementation, only the first element of $\tilde{\mathbf{u}}$ is implemented at the current interval and the remaining elements are discarded. This implementation can be expressed in a linear form by solving for the first control move, $\Delta u(t)$, in the control law. Because GPC is a subset of γGPC, the linear form of GPC will be a subset of the linear form of γGPC.

The first control element of (2.3.18) can be written as

$$\Delta u(t) = \mathbf{h}(\mathbf{w} - \mathbf{f}) + h_s(w_s - f_s) \tag{2.4.1}$$

where

$$\mathbf{h} = \text{ first row of } \left[\mathbf{G}^T \Gamma_y \mathbf{G} + \Lambda + \mathbf{G}_s^T \Gamma \mathbf{G}_s\right]^{-1} \mathbf{G}^T \Gamma_y \tag{2.4.2}$$

w is the setpoint vector, **f** is defined in (2.2.9), w_s is steady state setpoint and

$$h_s = \sum \left\{ \text{the first row of } \left[\mathbf{G}^T \Gamma_y \mathbf{G} + \Lambda + \mathbf{G}_s^T \Gamma \mathbf{G}_s \right]^{-1} \mathbf{G}_s^T \Gamma \right\} \tag{2.4.3}$$

$$f_s = \overline{G}_s \Delta u^f (t-1) + F_s y^f (t) \tag{2.4.4}$$

Expanding **w**, **f**, w_s and f_s in (2.4.1) and assuming a constant setpoint yields

$$\Delta u(t) = \begin{bmatrix} \mathbf{h} & h_s \end{bmatrix} \begin{bmatrix} w(t) - F_{N_1} y^f(t) - \overline{G}_{N_1} \Delta u^f (t-1) \\ w(t) - F_{N_1+1} y^f(t) - \overline{G}_{N_1+1} \Delta u^f (t-1) \\ \vdots \\ w(t) - F_{N_2} y^f(t) - \overline{G}_{N_2} \Delta u^f (t-1) \\ w(t) - F_s y^f(t) - \overline{G}_s \Delta u^f (t-1) \end{bmatrix} \tag{2.4.5}$$

Expanding (2.4.5) and grouping common $\Delta u(t)$, $w(t)$ and $y(t)$ terms gives

$$\left\{ C + q^{-1} \left[\sum_{j=N_1}^{N_2} \overline{G}_j h_j + \overline{G}_s h_s \right] \right\} \Delta u(t) = \left\{ C \left[\sum_{j=N_1}^{N_2} h_j + h_s \right] \right\} w(t) - \left\{ \sum_{j=N_1}^{N_2} F_j h_j + F_s h_s \right\} y(t) \tag{2.4.6}$$

where h_j is the j^{th} element of **h**. The above equation is now in the linear form

$$T \Delta u(t) = R w(t) - S y(t) \tag{2.4.7}$$

where

$$T = C + q^{-1} \left[\sum_{j=N_1}^{N_2} \overline{G}_j h_j + \overline{G}_s h_s \right] \tag{2.4.8}$$

$$R = C \left[\sum_{j=N_1}^{N_2} h_j + h_s \right] \tag{2.4.9}$$

$$S = \sum_{j=N_1}^{N_2} F_j h_j + F_s h_s \tag{2.4.10}$$

The order of the linear polynomials is:

$$nT = \max(nB, nC) \qquad nR = nC \qquad nS = \max(nA, nC\text{-}1) \tag{2.4.11}$$

2.5 The Discrete PID Control Law

Before the days of digital computers, continuous PID control of industrial processes was implemented using analog or pneumatic devices. In fact, pneumatic PID controllers are still manufactured and used today for local control of many remote processes. With the introduction of digital computers in the 1970's came the possibility of using advanced techniques such as LQG and minimum variance control (Åström and Wittenmark, 1973). Aerospace and defense industries used LQG control techniques extensively but process industries in general did not use advanced control extensively. Instead, industrial process control took the route of discretizing the continuous PID algorithm because it was already well understood and it provided adequate regulation of most processes. Subsequently, many *ad-hoc* and sub-optimal advancements were made for the PID control structure. The discrete PID controller can take many forms as shown in Åström and Wittenmark (1984) and Isermann (1981).

This section develops the incremental discrete PID control law from the ideal continuous algorithm given by

$$u(t) = K_{P_c} e(t) + K_{I_c} \int_0^t e(t)dt + K_{D_c} \frac{de(t)}{dt} \tag{2.5.1}$$

where *e(t)*, K_{P_c}, K_{I_c} and K_{D_c} are the error, non interacting proportional, integral and derivative constants, respectively. There are many variants of interacting PID control laws in industry. The non interacting algorithm (2.5.1) was chosen as a basis because it is the simplest PID form and interacting forms can be easily derived from it. A first order discretization of (2.5.1) results in the following discrete control law (in all subsequent development, the notation (t) denotes a discrete time variable)

$$u(t)=K_P e(t)+K_I \sum_{i=0}^{t} e(i)+K_D\left[e(t)-e(t-1)\right] \tag{2.5.2}$$

where

$$\begin{aligned} K_P &= K_{P_c} \\ K_I &= K_{I_c} T_S \\ K_D &= \frac{K_{D_c}}{T_S} \end{aligned} \tag{2.5.3}$$

and T_S is the sampling interval. Applying the differencing operator Δ to the controller output gives the incremental output.

$$\Delta u(t)=u(t)-u(t-1) \tag{2.5.4}$$

Substitution of the control law, (2.5.2) into (2.5.4) yields the incremental or velocity PID control law.

$$\Delta u(t)=\left[(K_P+K_I+K_D)+(-K_P-2K_D)q^{-1}+(K_D)q^{-2}\right]e(t) \tag{2.5.5}$$

A common industrial practice is to remove the setpoint signal from the derivative term in (2.5.5) to avoid abrupt control actions following a setpoint change otherwise known as the derivative kick. It is also not uncommon to remove the setpoint from the proportional term (Åström and Wittenmark, 1984) to further reduce large control actions following a setpoint change. The resulting setpoint on integral only (SP on I) controller is described by

$$\Delta u(t)=G_{Cw}w(t)-G_{Cy}y(t) \tag{2.5.6}$$

where

$$\begin{aligned} G_{Cw} &= K_I \\ G_{Cy} &= (K_P+K_I+K_D)+(-K_P-2K_D)q^{-1}+(K_D)q^{-2} \end{aligned} \tag{2.5.7}$$

Observation of the G_{Cw} and G_{Cy} polynomials in the SP on I control law (2.5.7), shows that a PI controller is first order in $y(t)$, a PID controller is second order in $y(t)$ and PI/PID controllers are zero order with respect to $w(t)$.

2.6 The PI/PID form of γGPC

As the published literature shows, there has been considerable interest over the years to incorporate process knowledge into PID tuning to improve control performance. Ziegler and Nichols (1942) developed a systematic method of PID controller tuning based on the period of sustained closed loop oscillations and the ultimate controller gain. From first order dynamics, the Cohen and Coon (1953) PID relations give ¼ amplitude decay response. PID tuning based on an integral of the error squared (ISE) criterion was developed by Lopez *et al.* (1967). Internal model control (IMC) techniques were used by Rivera *et al.* (1986) for PID controller design. An excellent discussion on PID tuning using time and frequency domain techniques can be found in Åström and Häggland (1988). All of the above PID tuning techniques are based on some *ad-hoc* criterion which may work in some cases but may often fail in other cases. Noise variance and disturbance correlation are important considerations for controller tuning, although, the PID tuning methods mentioned above do not compensate for noise.

It was recognized by McIntosh (1988) and Henningsen *et al.* (1990) that a discrete PID control law such as (2.5.6) is structurally equivalent to standard GPC given some restrictions to the ARIMAX model on which GPC is based. This section analyzes the conditions under which γGPC is equivalent to PID. A predictive PID control law that is based on a long range predictive control strategy will prove to be an excellent candidate for the replacement of classical PID. Consider the deterministic ($C = 1$) linear γGPC control law for a second order A polynomial and zero order B polynomial which is a second order plant model without time delay. From (2.4.11), the linear polynomials T, R and S are of order 0, 0 and 2, respectively so the control law (2.4.7) can be written as

$$\Delta u(t) = r_0 w(t) - \left(s_0 + s_1 q^{-1} + s_2 q^{-2}\right) y(t) \tag{2.6.1}$$

where r_i and s_i are the coefficients of the R and S polynomials, respectively. Recall that the T polynomial has a leading 1, therefore, a zero order T polynomial is unity. Equating the PID SP on I control law (2.5.6) and the linear γGPC control law (2.4.7) yields an exact match if

$$G^d_{Cw} = R \tag{2.6.2}$$

$$G^d_{Cy} = S \tag{2.6.3}$$

The superscript in (2.6.2) and (2.6.3) indicates a deterministic model representation. The PID tuning constants in (2.5.7) can be expressed in terms of the linear γGPC coefficients by equating (2.5.7) with (2.6.1) to yield

$$\begin{aligned} K_P &= -\left(s_1 + 2s_2\right) = s_0 - r_0 - s_2 \\ K_I &= r_0 \\ K_D &= s_2 \end{aligned} \tag{2.6.4}$$

A first order plant model results in a first order S polynomial and an equivalent PI controller while a second order plant yields a PID controller from the relations in (2.6.4). The block diagram of the predictive PID control loop in Figure 2.1 reveals a two degree of freedom PID controller defined by (2.5.6) (*dist* and *y* denote continuous time variables).

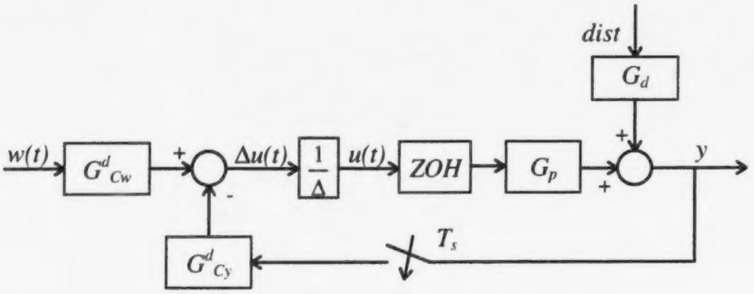


Figure 2.1: Block diagram of the predictive deterministic PID control loop.

A comparison of GPC and equivalent PID servo and regulatory response is shown in Figure 2.2 for the unstable plant $G_P = \frac{-0.015}{1 - 1.95q^{-1} + 0.935q^{-2}}$. A step disturbance of 0.1 is applied/removed at 100/150 and 250/300 sampling instants, respectively. The simulation shows an exact match between GPC and PID and the control performance is satisfactory for a difficult control problem.

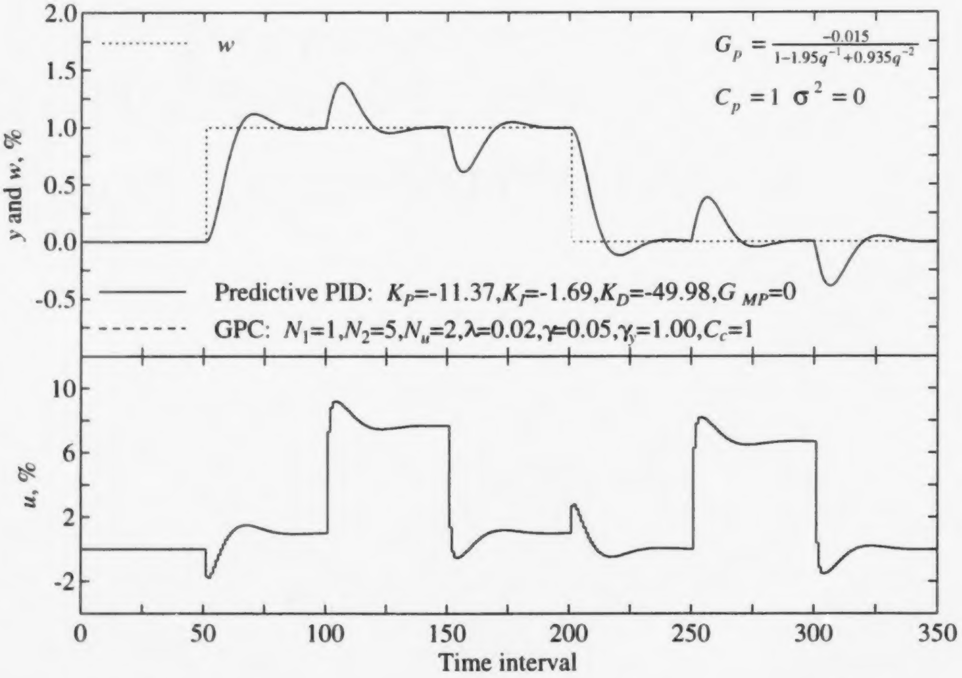


Figure 2.2: GPC and equivalent PID control response for a discrete second order plant.

The following remarks summarize the main results for this section. A long range predictive PI and PID control law results from equating the linear polynomials in GPC with the PID constants given by the relation (2.6.2) for first and second order plants, respectively. This implies that the proportional and derivative constants in the chosen PID form are predictive in a long range sense when chosen optimally. The B polynomial is restricted to zero order and the C polynomial is restricted to 1 although there are no restrictions on the γGPC tuning parameters N_1, N_2, N_u, λ, γ and γ_y. These restrictions imply that the PID controller cannot be optimal for plant models with higher orders, non zero time delays or stochastic disturbances.

2.7 The Multistep Long Range Predictor, G_{MP}

2.7.1 Time Delay Compensation

Although the PID controller proposed in section 2.5 is interesting, it does not offer a practical solution to control problems with time delays or realizable second order plant models (i.e. with two coefficients in the numerator). Almost all chemical processes contain time delays and higher order dynamics, therefore, an effective control law must account for such adversities. The conventional PID structure can only deal with a time delay and higher order dynamics by detuning which can result in mediocre or unacceptable performance. As the control problem becomes more difficult, conventional PID is less capable of achieving the control objective regardless which method of choosing the PID parameters is used. Smith (1957) proposed a method for time delay compensation by using an internal model that consists of the difference of the plant model and the undelayed plant model. Under perfect modelling conditions, (a utopian concept) the Smith predictor removes the time delay element from the characteristic equation which allows more aggressive PID tuning. The major disadvantage of the Smith predictor is it's sensitivity to time delay mismatch because it gives a k step ahead prediction where k is the time delay plus one. Walgama (1986) found that an *ad-hoc* exponential filter in the feedback path of a PID controller with a Smith predictor gives improved robustness in the case of a time delay mismatch. It was shown by Harris *et al.* (1982) that a minimum variance controller (MVC) reduces to a PI control structure with a Smith predictor for first order plant models with a time delay.

The practical extension of MVC is to a long range multistep prediction as in GPC. A significant advantage of a long range prediction over a k step ahead prediction is reduced sensitivity to model-plant-mismatch. In essence, the LRPC strategy plots a smooth trajectory over the prediction horizon to guide the controlled variable to the setpoint. In comparison, the MVC controller takes the control action necessary to bring the controlled variable to the setpoint after k steps. This implies that a long range weighted predictor must be developed so that the model based PID controller is equivalent to GPC. The objective of this section is to develop a multistep long range weighted predictor by comparison with GPC.

The deterministic linear GPC control law (2.4.6) can be expressed by

$$\left\{1+q^{-1}\left[\sum_{j=N_1}^{N_2}\overline{G}_j h_j+\overline{G}_s h_s\right]\right\}\Delta u(t)=\left[\sum_{j=N_1}^{N_2}h_j+h_s\right]w(t)-\left[\sum_{j=N_1}^{N_2}F_j h_j+F_s h_s\right]y(t) \qquad (2.7.1)$$

which can be rewritten as

$$\Delta u(t)=\left[\sum_{j=N_1}^{N_2}h_j+h_s\right]w(t)-\left[\sum_{j=N_1}^{N_2}F_j h_j+F_s h_s\right]y(t)-\left[\sum_{j=N_1}^{N_2}\overline{G}_j h_j+\overline{G}_s h_s\right]\Delta u(t-1) \qquad (2.7.2)$$

From section 2.5, the linear γGPC polynomials R and S are equivalent to the discrete PID terms G_{Cw}^{d} and G_{Cy}^{d}, respectively. The discrete PID terms can then be substituted into (2.7.2) as follows

$$\Delta u(t)=G_{Cw}^{d}w(t)-G_{Cy}^{d}y(t)-\left[\sum_{j=N_1}^{N_2}\overline{G}_j h_j+\overline{G}_s h_s\right]\Delta u(t-1) \qquad (2.7.3)$$

Because (2.7.2) is mathematically equivalent to (2.7.3), it is evident that the term

$$\left[\sum_{j=N_1}^{N_2}h_j\overline{G}_j+\overline{G}_s h_s\right]\Delta u(t-1) \qquad (2.7.4)$$

will be a predictor when used in addition to the discrete PID control law. The GPC law is an optimal multistep predictive control law which implies that (2.7.4) is an optimal multistep weighted predictor when used as an internal model for PID control. The deterministic predictor is defined as

$$G_{MP}^{d}=\frac{\sum_{j=N_1}^{N_2}h_j\overline{G}_j+\overline{G}_s h_s}{G_{Cy}} \qquad (2.7.5)$$

where the superscript, *d*, indicates a deterministic plant. The incremental model based PID algorithm equivalent to (2.7.1) can now be written as

$$\Delta u(t)=G^d_{Cw}w(t)-G^d_{Cy}y(t)-G^d_{Cy}G^d_{MP}\Delta u(t-1) \qquad (2.7.6)$$

where the PID constants are determined from the relations in (2.6.4). The model order is restricted to a maximum of two because the maximum order of G^d_{Cy} is two. There are no restrictions on the choice of γGPC tuning parameters. The block diagram of the model based predictive PID controller can be expressed as shown in Figure 2.3 which is equivalent to Figure 2.1 in addition to the predictor $G^d{}_{MP}$.

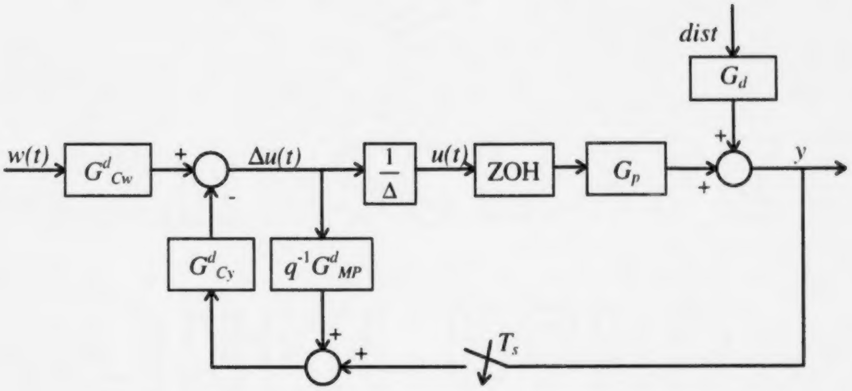


Figure 2.3: Block diagram of the predictive model based PID control loop.

The control performance of GPC and equivalent predictive PID is demonstrated in Figure 2.4 for the second order plant: $G_p = \frac{e^{-5s}}{(3s+1)(5s+1)}$. A step disturbance of 0.05 is applied/removed at 100/150 and 250/300 sampling instants, respectively. The simulation results in Figure 2.4 show excellent performance for the predictive PID scheme for a process with a significant time delay.

The main result of this section is a PID algorithm with a multistep predictive internal model that is equivalent to GPC. This provides multistep weighted time delay compensation although the *C* polynomial is restricted to unity (i.e. the plant is represented by an ARX or an ARIX model) and the maximum plant model order is two.

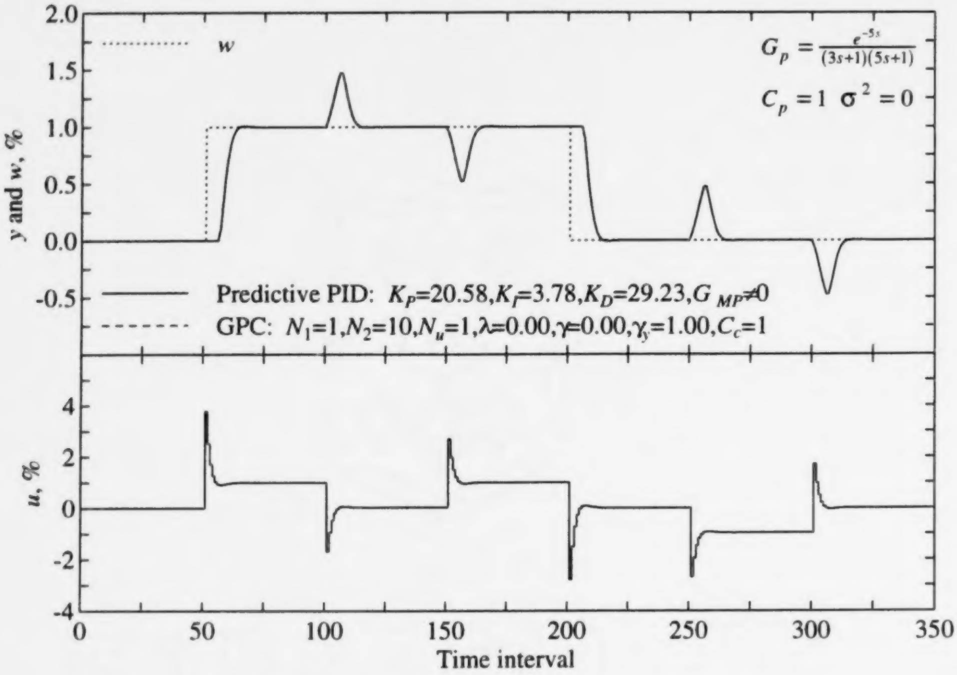


Figure 2.4: GPC and equivalent PID response to a second order plant.

2.7.2 Extension of G_{MP} to Stochastic Disturbance Compensation

The design relations for determining PID constants discussed in section 2.7.1 are based on noise free deterministic plants. However, real processes are subject to random disturbances which may be correlated. A PID controller must be significantly detuned (particularly the derivative term) to provide acceptable performance when significant noise is present. Moreover, the detuning of a PID controller in the presence of process noise is usually done in some *ad-hoc* fashion. GPC compensates for correlated noise by including the noise model in the development of the control law. The moving average term, $C/A\Delta$, in the GPC control law provides the basis for a realistic noise model. A PID control law based on GPC with a non unity C polynomial will not require *ad-hoc* detuning for noisy plants because the noise model is included in the control law. In addition, the control performance will be improved because the control law predicts the effect of past correlated disturbances rather than simply reacting to feedback.

This section extends the predictor, G_{MP}, to include stochastic disturbance compensation. It is recommended by McIntosh *et al.* (1991) that the C polynomial be used as a tuning parameter rather than an approximation of C_p. In the following sequel, the notation, C_c is used to indicate the use of C as a controller tuning parameter. The linear γGPC control law (2.4.6) can be expressed as

$$\begin{aligned}\Delta u(t)=&\left\{C_c\left[\sum_{j=N_1}^{N_2}h_j+h_s\right]\right\}w(t)-\left\{\sum_{j=N_1}^{N_2}F_jh_j+F_sh_s\right\}y(t)\\&-\left\{\left[\sum_{j=N_1}^{N_2}\overline{G}_jh_j+\overline{G}_sh_s\right]+\left(C_c-1\right)\right\}\Delta u(t-1)\end{aligned}\tag{2.7.7}$$

because there is always a leading 1 in the C_c polynomial. The first or the servo term in (2.7.7) is of order nC_c which does not fit into the PID SP on I form unless C_c=1. In industry, most processes are operated in a regulatory fashion for the vast majority of the time and changes in setpoint are infrequent. A zero order approximation of the servo term in (2.7.7) would compromise the servo performance but leave regulatory control unchanged. The most logical zero order approximation is the final or steady state value of the C_c polynomial in the servo term of (2.7.7). A steady state approximation of C_c will always give a conservative or detuned servo response but will be offset free during regulatory response. The following expression is such an approximation to (2.7.7)

$$\begin{aligned}\Delta u(t)=&\left\{\sum_{j=1}^{nC_c}C_{c\,j}\left[\sum_{j=N_1}^{N_2}h_j+h_s\right]\right\}w(t)-\left\{\sum_{j=N_1}^{N_2}F_jh_j+F_sh_s\right\}y(t)\\&-\left\{\left[\sum_{j=N_1}^{N_2}\overline{G}_jh_j+\overline{G}_sh_s\right]+\left(C_c-1\right)\right\}\Delta u(t-1)\end{aligned}\tag{2.7.8}$$

The PID equivalent to (2.7.8) is given by

$$\Delta u(t)=G_{Cw}w(t)-G_{Cy}y(t)-G_{MP}G_{Cy}\Delta u(t-1)\tag{2.7.9}$$

where

$$G_{Cw}=\sum_{j=1}^{nC_c}C_{c\,j}\left[\sum_{j=N_1}^{N_2}h_j+h_s\right]\tag{2.7.10}$$

$$G_{Cy}=\sum_{j=N_1}^{N_2}F_jh_j+F_sh_s\tag{2.7.11}$$

$$G_{MP} = \frac{\left[\sum_{j=N_1}^{N_2} \overline{G}_j h_j + \overline{G}_s h_s\right] + (C_c - 1)}{G_{Cy}} \tag{2.7.12}$$

The PID controller constants K_P, K_I and K_D can be solved for in the same manner as in section 2.5 because the order of (2.7.10) and (2.7.11) are 0 and nA, respectively. The stochastic PID polynomials in (2.7.9) are equivalent to the deterministic PID polynomials when $C_c = 1$, therefore, the deterministic case is naturally a subset of the stochastic case. Figure 2.5 shows the block diagram of the stochastic predictive PID control loop. The stochastic predictive PID controller can also be expressed in a model based PID form as shown in Figure 2.6. This alternative representation uses all three control modes (P, I and D) on setpoint changes which differs from the SP on I form discussed previously. The setpoint filter in Figure 2.6 has the same effect as removing the proportional and derivative action from setpoint changes. Industrial implementation of predictive PID may be better suited to the representation in Figure 2.5 or Figure 2.6 depending on the limitations of the specific control computer. Other representations of predictive PID can be developed by simple block diagram manipulations to Figure 2.5.

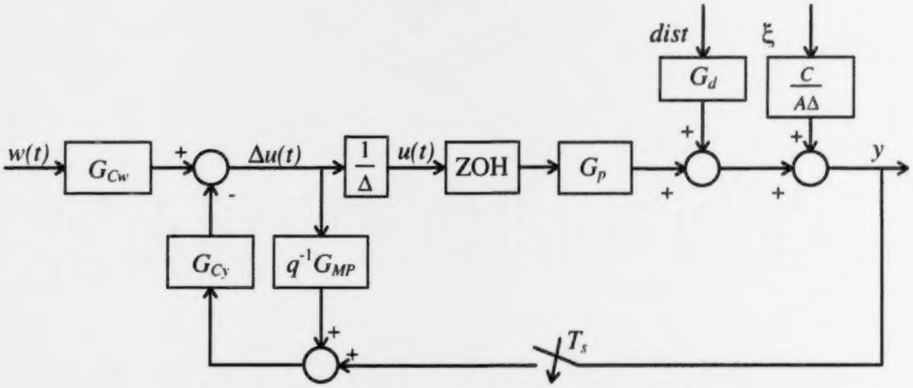


Figure 2.5: Block diagram of the stochastic predictive PID control loop.

Figure 2.7 shows the performance of a deterministic PID controller given by (2.7.9) for the plant model: $G_p = \frac{(-2s+1)}{(6s+1)(2s+1)}$. The plant noise is correlated by $C_p = 1 - 0.8q^{-1}$ with variance $\sigma^2 = 0.0005$. The controlled variable response is satisfactory although the manipulated variable variance is excessive. Figure 2.8 shows that the response of the stochastic PID controller is slightly more sluggish than GPC for servo response, however, regulatory response is

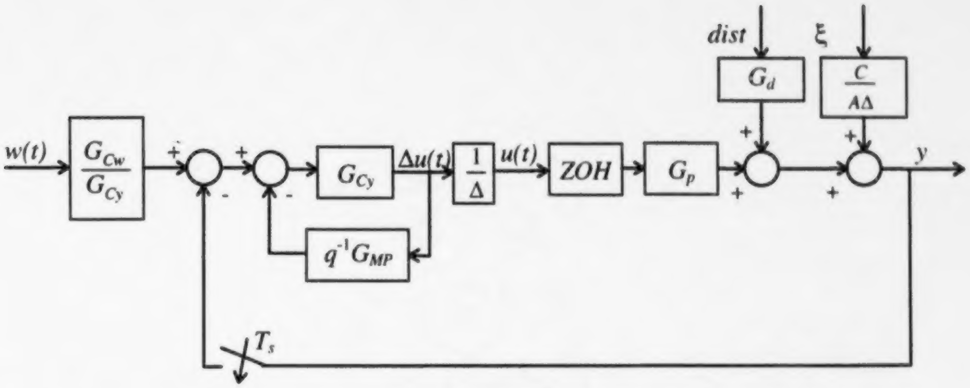


Figure 2.6: Alternate block diagram of the stochastic predictive PID control loop.

equivalent. The controller output variance in Figure 2.8 is significantly smaller than the controller output variance in Figure 2.7. A stochastic control law compensates for correlated noise which results in less aggressive control. The detuned PID servo response compared with the GPC control performance presented in Figure 2.8 is expected because of the approximation of the C_c polynomial in G_{Cw}. This detuned response only occurs for a short period following setpoint changes. This should not be a problem for most processes because regulation is the typical control objective. The amount of detuned servo response is proportional to the final value of C_c (i.e. $C_c(1)$). If $C_c(1)$ is very close to 0 then the servo response of predictive PID will be very detuned. As the final value of C_c approaches zero, the servo approximation of predictive PID, (2.7.10), deviates further from the GPC solution. This is illustrated in Figure 2.9 for the same plant model and controller tuning as in Figure 2.8 except C_c is now set to $1\text{-}0.92q^{-1}$. The servo response for the predictive PID controller is significantly slower than achieved using the GPC law as shown by the results given in Figure 2.9. Continuous processes do not usually require frequent setpoint changes. Furthermore, it may be desirable to change from one steady state to another slowly to avoid upsets in downstream and parallel processes. An *ad-hoc* solution to increase servo aggression in the predictive PID controller was considered for the case where the detuned servo response is unacceptable. This method involves including the proportional action for setpoint changes (PI on SP) for the predictive PID scheme developed earlier. Figure 2.10 shows the response of the PI on SP form of the predictive PID controller for the same plant and controller tuning as in Figure 2.9. The servo response of the PID controller is now almost identical to the GPC response. However, this solution should only be considered for cases where $C_c(1)$ is very close to zero and the servo response is too slow. The stability properties of the

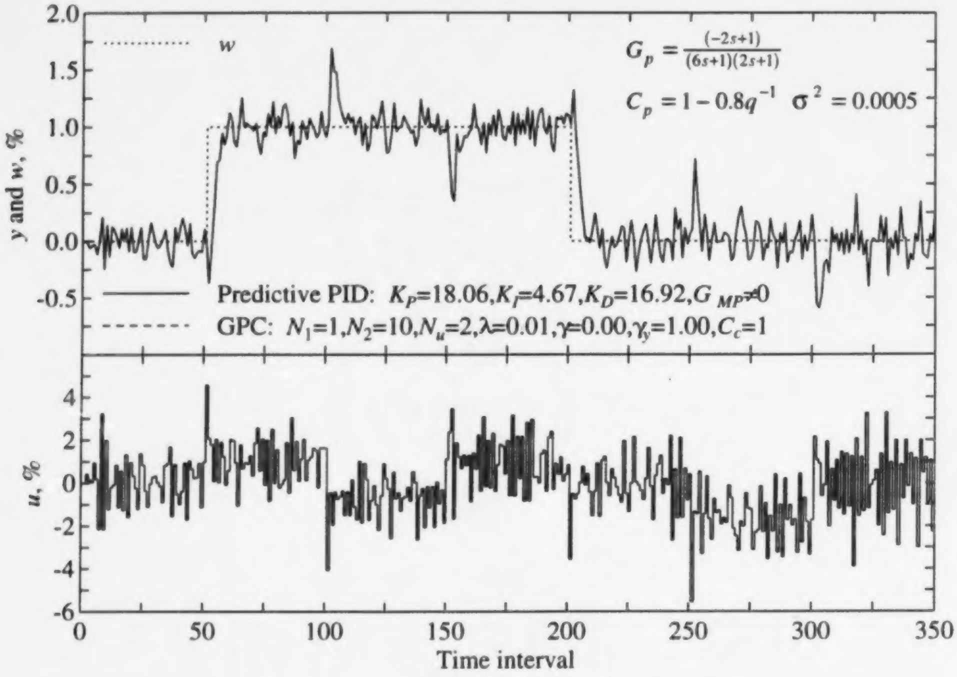


Figure 2.7: Response of deterministic GPC and predictive PID to a stochastic process.

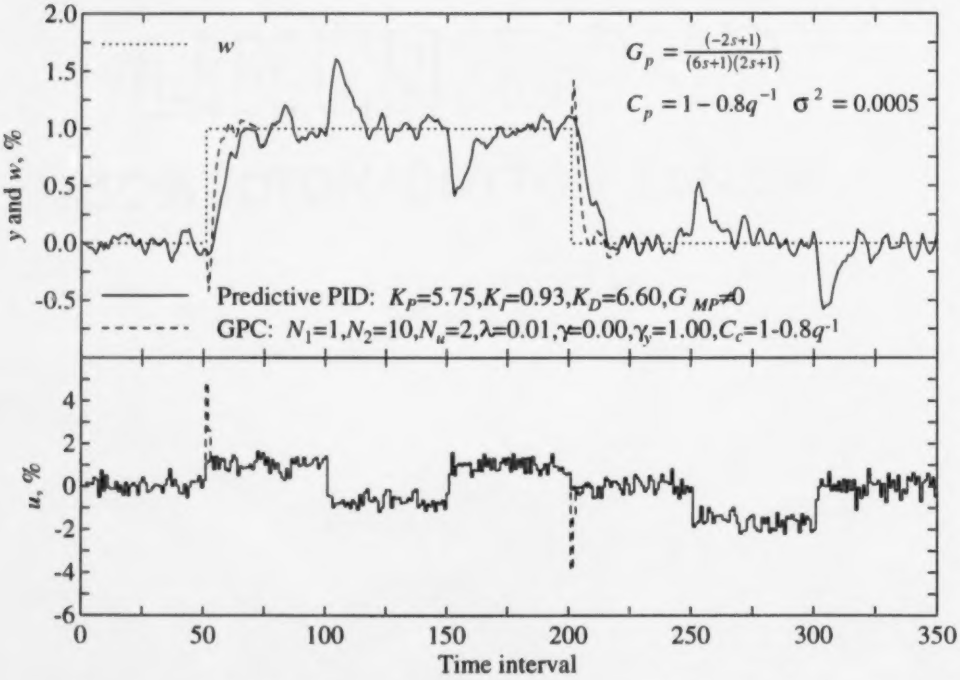


Figure 2.8: Response of stochastic GPC and predictive PID to a stochastic process.

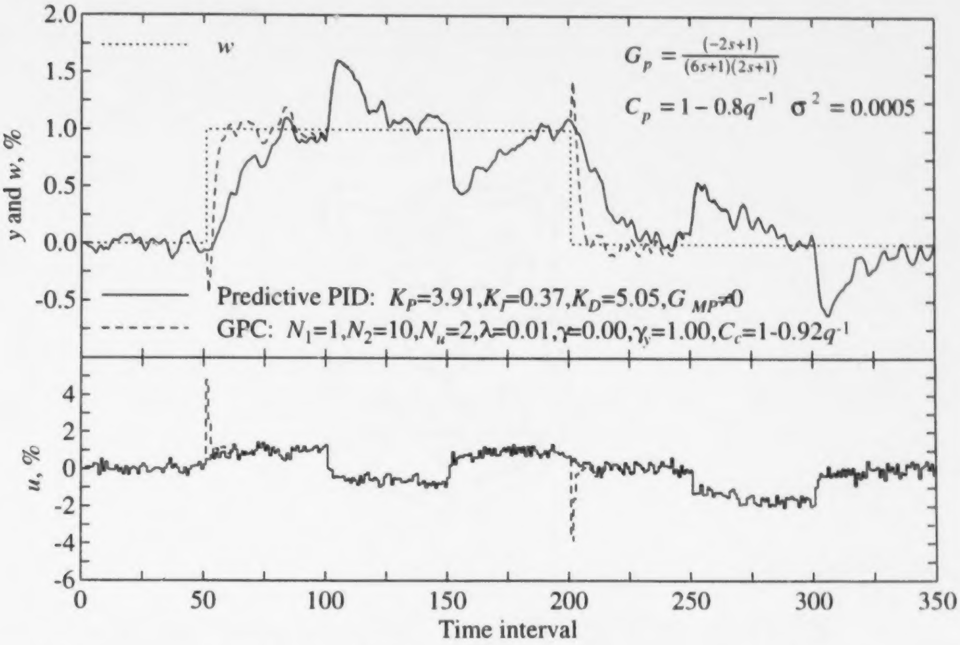


Figure 2.9: Response of GPC and predictive PID with an over estimated C polynomial to a stochastic process.

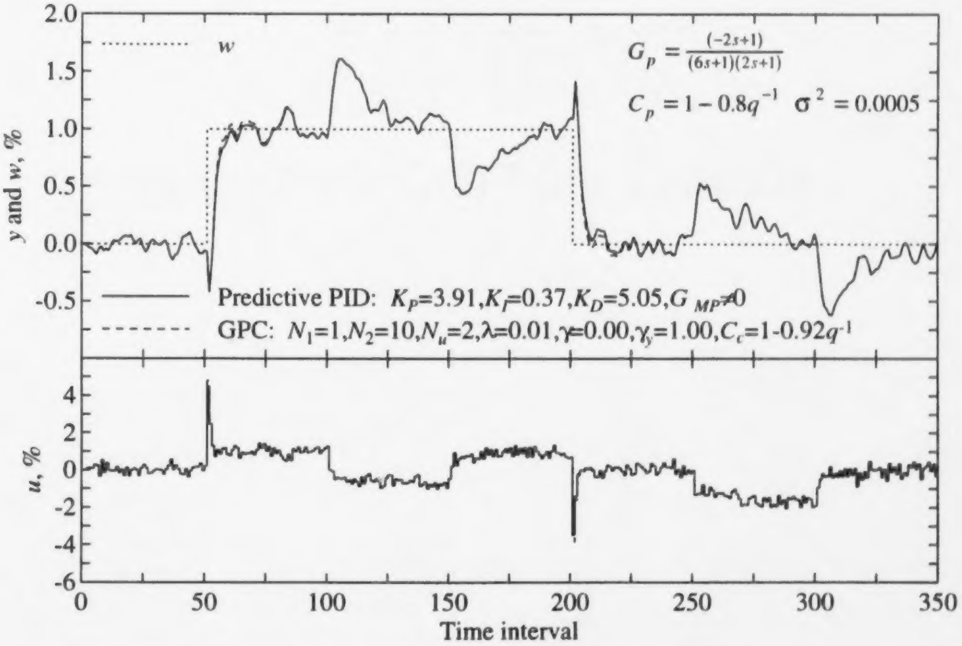


Figure 2.10: Response of GPC and predictive PID with a PI on SP form and an over estimated C polynomial to a stochastic process.

predictive PID scheme can not be guaranteed when a PI on SP form is used. Figure 2.11 shows the response of the predictive PID scheme with a PI on SP form and $C_c = 1$ for the same plant as in Figure 2.7. As can be seen, the result of employing the predictive PI on SP law for this case is a large unacceptable change in controller output following setpoint changes.

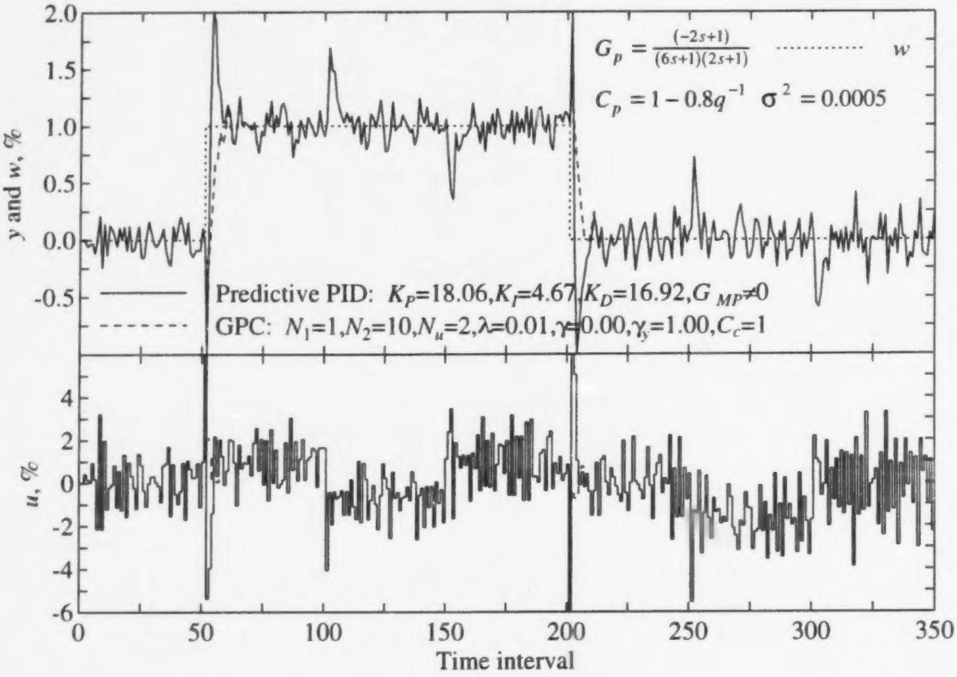


Figure 2.11: Response of GPC and predictive PID with a PI on SP form to a stochastic process.

2.8 Interpretations of the Predictor, G_{MP}

It is apparent from the previous sections that a discrete PID control law with the internal model G_{MP} is equivalent to γGPC. Conceptually, G_{MP} can be interpreted as a long range predictor by comparison with γGPC. The objective of this section is to gain a deeper understanding of G_{MP} by further analysis. Consider the special case where γ is set to zero and $C_c = 1$. The resulting model based PID controller can be interpreted by expanding the G_{MP} term. Recall from (2.2.4) that $\overline{G}_j$ contains the nB-1 j^{th} step ahead step response coefficients for a deterministic plant model. This can be illustrated by the following example. Let the first order plus time delay plant be given by $A = 1 - 0.9q^{-1}$ and $B = 0.1q^{-1}$ (note that the plant ZOH is removed from the model). The step response coefficients are: 0.0, 0.100, 0.190, 0.271, 0.344, ..., 1.000. For the first step (j = 1) $\overline{G}_1 = 0.1$ and for the second step (j = 2) $\overline{G}_2 = 0.19$, therefore, $\overline{G}_j$ contains the j^{th} step ahead prediction and is of order nB-1. The predictor G_{MP} can now be interpreted as an optimal weighted sum of j step ahead predictions from the minimum prediction horizon, N_1 to the maximum prediction horizon, N_2. This is illustrated in Figure 2.12. The h_j weight in Figure 2.12 corresponds to the j^{th} coefficient of **h** where

$$\mathbf{h} = \text{first row of } \left[\mathbf{G}^T\Gamma_y\mathbf{G}+\Lambda\right]^{-1}\mathbf{G}^T\Gamma_y$$

The block diagram in Figure 2.12 can be extended to include γ weighting by including the $\overline{G}_s$ and h_s terms. The interpretation of G_{MP} is more difficult to see when the plant model is stochastic i.e. $C_c \neq 1$. Both Diophantine identities (2.2.2) and (2.2.5) contain the C_c polynomial from which the linear γGPC polynomials are based. Therefore, the extension of the above interpretation to include stochastic disturbance compensation is not as simple as inserting a $\frac{1}{C}$ filter in the block diagram. All of the linear PID polynomials, G_{Cw}, G_{Cy} and G_{MP} are strong functions of C_c. The C_c polynomial has the effect of detuning all of the linear PID polynomials. Extending the block diagram to include $C_c \neq 1$, and γ weighting results in Figure 2.13.

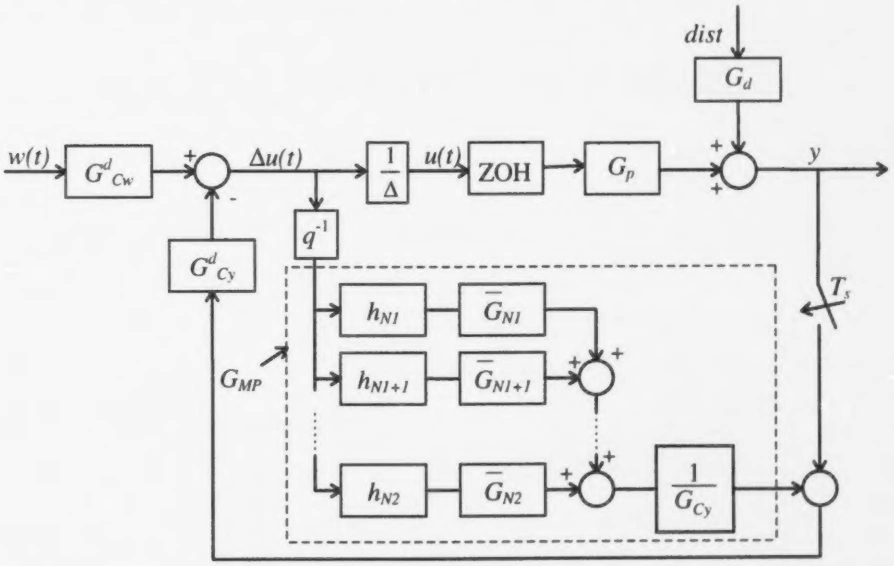


Figure 2.12: Block diagram of deterministic predictive PID control loop with the expanded predictor, G_{MP}.

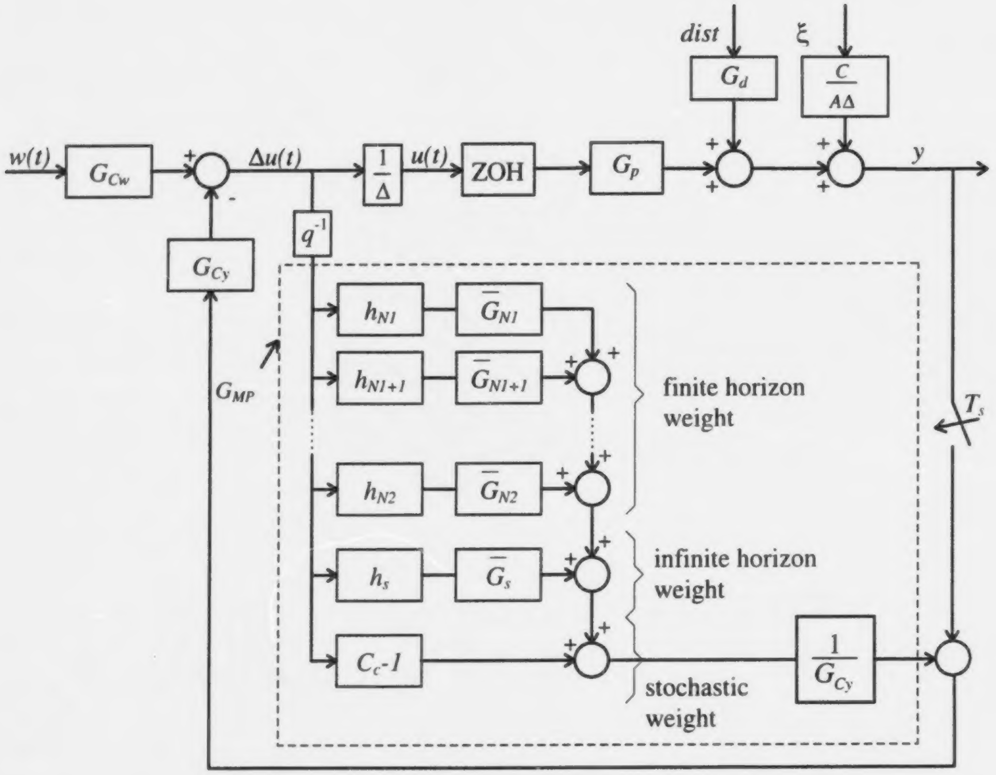

dist
ξ
G_d
$\frac{C}{A\Delta}$
w(t)
G_{Cw}
+
-
Δu(t)
$\frac{1}{\Delta}$
u(t)
ZOH
G_p
+
+
+
+
y
q^{-1}
G_{Cy}
G_{MP}
h_{N1}
$\overline{G}_{N1}$
h_{N1+1}
$\overline{G}_{N1+1}$
finite horizon weight
T_s
h_{N2}
$\overline{G}_{N2}$
h_s
$\overline{G}_s$
infinite horizon weight
C_c-1
stochastic weight
$\frac{1}{G_{Cy}}$

Figure 2.13: Block diagram of the stochastic predictive PID control loop with the expanded predictor, G_{MP}.

2.9 Infinite Horizon GPC and PID

In practical applications of GPC, it is recommended that a maximum prediction horizon, N_2, corresponding to 50% to 90% of the rise time of the process be used (Kwok and Shah, 1994). For a stable overdamped process, most of the dynamics are contained in the first time constant. Unstable control response may result from an N_2 that is too short. Setting N_2 to infinity (∞GPC) results in mean level control or steady state model inverse control (Clarke, 1987b; McIntosh, 1991). The closed loop poles of a mean level controller are the same as the open loop poles, (Clarke *et al.*, 1987b), therefore, for an open loop stable process without model plant mismatch, mean level control response is guaranteed to be stable. It is generally desirable to set the closed loop poles to be somewhat faster than the open loop poles although robustness to model-plant-mismatch must be compromised for gains in performance. Mean level control offers a conservative but robust approach to automatic control. Clarke (1991) states that the mean level equivalent in GPC is often very effective in difficult control situations. It was verified experimentally by Lambert (1987) that the disturbance rejection abilities of mean level GPC are excellent.

This section analyzes the conditions under which γGPC and PID are equivalent to mean level control. Setting N_2 to infinity in GPC is not a practical implementation, however, the equivalent in γGPC can be accomplished by removing the finite horizon weight i.e. $\gamma_y = 0$. The following theorem formally shows this result.

Theorem 1

For an open loop stable process, the GPC control law resulting from the minimization of J (2.3.15) with $N_u = 1$, $N_2 \to \infty$ and $\lambda = 0$ (∞GPC) is equivalent to the γGPC control law where $N_u = 1$, $\gamma = 1$, $\gamma_y = 0$ and $\lambda = 0$ (C is the same for both control schemes).

Proof

Expanding the **h** term in the linear form of standard GPC ($\gamma = 0$, $\gamma_y = 1$) with $N_u = 1$, $N_2 \to \infty$ and $\lambda=0$ gives

$$\mathbf{h} = \left[\mathbf{G}^T\mathbf{G}\right]^{-1}\mathbf{G}^T$$
$$= \left[\sum_{j=1}^{\infty} g_j^2\right]^{-1} \left[g_1 \quad g_2 \quad \cdots \quad g_\infty\right]$$

where g_j and g_∞ are the step response coefficients and the steady state gain, respectively.

For a stable plant model, $\sum_{j=1}^{\infty} h_j$, $\overline{G}_j$ and F_j converge to $\frac{1}{g_\infty}$, $\overline{G}_\infty$ and F_∞, respectively. The linear control law can now be expressed as

$$\left[C_c + q^{-1}\frac{\overline{G}_\infty}{g_\infty}\right]\Delta u(t) = \frac{C_c}{g_\infty}w(t) - \frac{F_\infty}{g_\infty}y(t) \qquad (2.9.1)$$

The linear form of γGPC with $N_u = 1$, $\gamma = 1$ and $\gamma_y = 0$ is given by

$$\left[C_c + q^{-1}\frac{\overline{G}_s}{g_s}\right]\Delta u(t) = \frac{C_c}{g_s}w(t) - \frac{F_s}{g_s}y(t) \qquad (2.9.2)$$

but from section 2.2

$$\begin{aligned}\overline{G}_\infty &= \overline{G}_s\\ F_\infty &= F_s\\ g_s &= \frac{B(1)}{A(1)}\end{aligned}$$

so (2.9.1) is equivalent to (2.9.2).

A PID equivalent to ∞GPC, denoted as ∞PID, can be developed following the same procedure as utilized in section 2.5 and 2.6. From (2.9.2), the linear polynomials for ∞GPC are

$$\begin{aligned}T &= C_c + q^{-1}\frac{\overline{G}_s}{g_s}\\ R &= \frac{C_c}{g_s}\\ S &= \frac{F_s}{g_s}\end{aligned} \tag{2.9.3}$$

Equating the PID polynomials in (2.7.9) with (2.9.3) along the same lines of section 2.6 yields

$$\begin{aligned}G_{Cw}^{\infty} &= \frac{C_c(1)}{g_s}\\ G_{Cy}^{\infty} &= \frac{F_s}{g_s}\\ G_{MP}^{\infty} &= \frac{\overline{G}_s + g_s(C_c - 1)}{F_s}\end{aligned} \tag{2.9.4}$$

The infinite prediction horizon PID control law is then given by

$$\Delta u(t) = G_{Cw}^{\infty} w(t) - G_{Cy}^{\infty} y(t) - G_{MP}^{\infty} G_{Cy}^{\infty} \Delta u(t-1) \tag{2.9.5}$$

The terms in (2.9.4) are significantly simpler than the finite horizon case that was developed in sections 2.6 and 2.7. Equation (2.9.4) can be further decomposed by substitution of (2.3.8), (2.3.9), (2.3.10) and (2.3.11) to give

$$
\begin{aligned}
G^{\infty}_{Cw} &= \frac{C_c(1)A(1)}{B(1)} \\
G^{\infty}_{Cy} &= \frac{C_c(1)}{B(1)}A \\
G^{\infty}_{MP} &= \frac{B(1)C_c - C_c(1)B + B(1)\Delta(C_c - 1)}{C_c(1)A\Delta}
\end{aligned}
\qquad (2.9.6)
$$

The above polynomials now are independent of any GPC tuning parameters. Predictive PID controller constants, K_P, K_I and K_D can be expressed in terms of the model parameters by comparing the coefficients of (2.9.6) with the linear relations in (2.5.7) which results in

$$
\begin{aligned}
K_P &= \frac{(a_1 + 2a_2)(-C_c(1))}{B(1)} \\
K_I &= \frac{C_c(1)A(1)}{B(1)} \\
K_D &= \frac{C_c(1)}{B(1)}a_2
\end{aligned}
\qquad (2.9.7)
$$

Figure 2.14 shows the response of the ∞PID controller for the same plant and disturbance that gave rise to the response presented in Figure 2.8. Although the ∞PID control law is very simple, the control performance is quite acceptable providing a very similar disturbance rejection trajectory compared to the control response of the significantly more complicated GPC law shown in Figure 2.8. However, the servo performance of ∞PID is considerably slower than the GPC response in Figure 2.8 which is expected for a mean level controller. Increasing servo aggression on the ∞PID controller is accomplished simply by including the proportional term to setpoint changes as discussed in section 2.7.

The ∞PID controller defined by (2.9.5) no longer requires the solution of the Diophantine identities as was the case in the predictive PID controller developed in section 2.6. This has several advantages. Implementation of this control law is very simple and should not require highly specialized control engineers. Also, the understanding of the control law can be made entirely in terms of the steady state predictions rather than a weighted sum of finite horizon predictions. This should be significantly easier to understand than standard GPC or DMC for operations personnel that are accustomed to PID control. The fact that the Diophantine identities are not required also means that the execution of the ∞PID controller is very efficient in

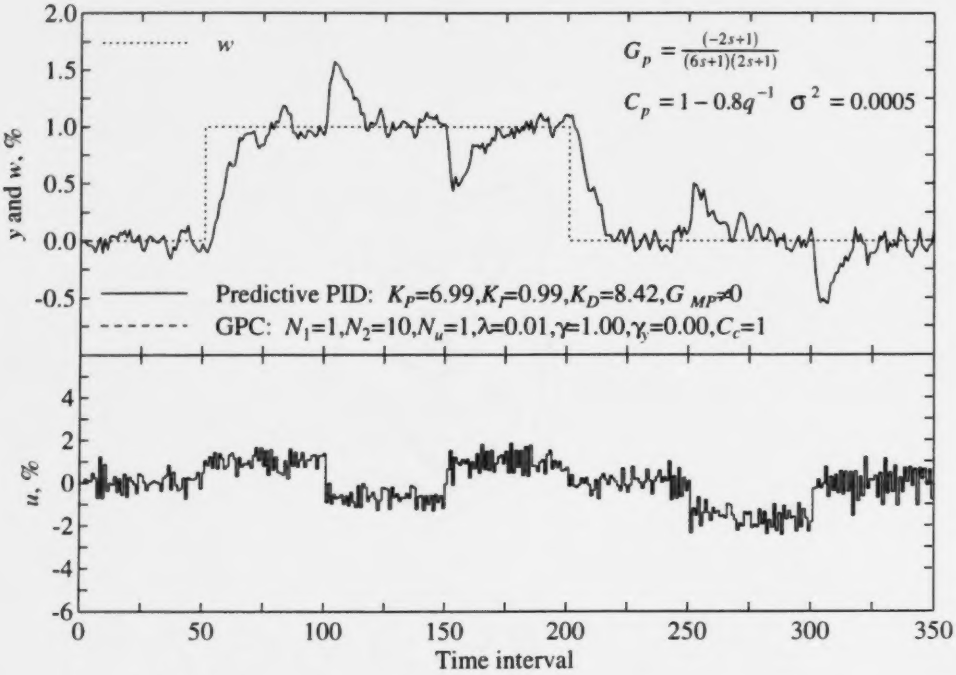


Figure 2.14: response of ∞GPC and ∞PID to a stochastic plant.

comparison to GPC or DMC. The adaptive implementation will also be very efficient because only simple multiplications are required to compute the control law. The relations in (2.9.6) are only linked to GPC by the ARIMAX model polynomials.

When the C polynomial is used as a tuning parameter rather than an estimation of the plant, the ∞PID control law can be considered to be a robust model based controller with two tuning parameters. It is tempting to choose a C_c polynomial that is not physically realizable in order to increase controller aggression for the mean level case. If $C_c = 1+cq^{-1}$, it is not physically possible for c to be greater than zero because current measured values are not correlated to future noise. Some intuitive reasoning may suggest using a positive c to increase controller aggression because a negative value of c detunes the controller response. However, by considering c as a tuning parameter and setting it to a value greater than zero, the response may be degraded rather than improved which is illustrated in Figure 2.15 for $c = 0.4$. Therefore, use of positive values of c in order to increase controller aggression is not recommended.

Use of the C polynomial as a tuning parameter in ∞PID can be analyzed more rigorously by observing the closed loop characteristic equation as C_c changes. The closed loop characteristic equation for γGPC as given by (Kwok and Shah, 1994) is

$$CE = A\Delta T + q^{-1}BS \tag{2.9.8}$$

which is also equivalent to the characteristic equation for predictive PID. For ∞PID, T and S are defined by (2.9.3). Substitution of (2.3.10), (2.3.11) and (2.9.3) into (2.9.8) for the infinite horizon configuration yields the following characteristic equation

$$CE_{\infty PID} = AC_c \tag{2.9.10}$$

If C_c=1 in (2.9.10), then the closed loop poles are in the same position as the open loop poles which is the mean level case. Construction of a root locus of ∞PID is now a trivial matter of plotting the open loop poles and following the trajectory of the roots of the C_c polynomial. It is also clear that selecting a positive value of c will place the pole in the left half of the z plane unit circle. This reaffirms the recommendation of not selecting positive values of c.

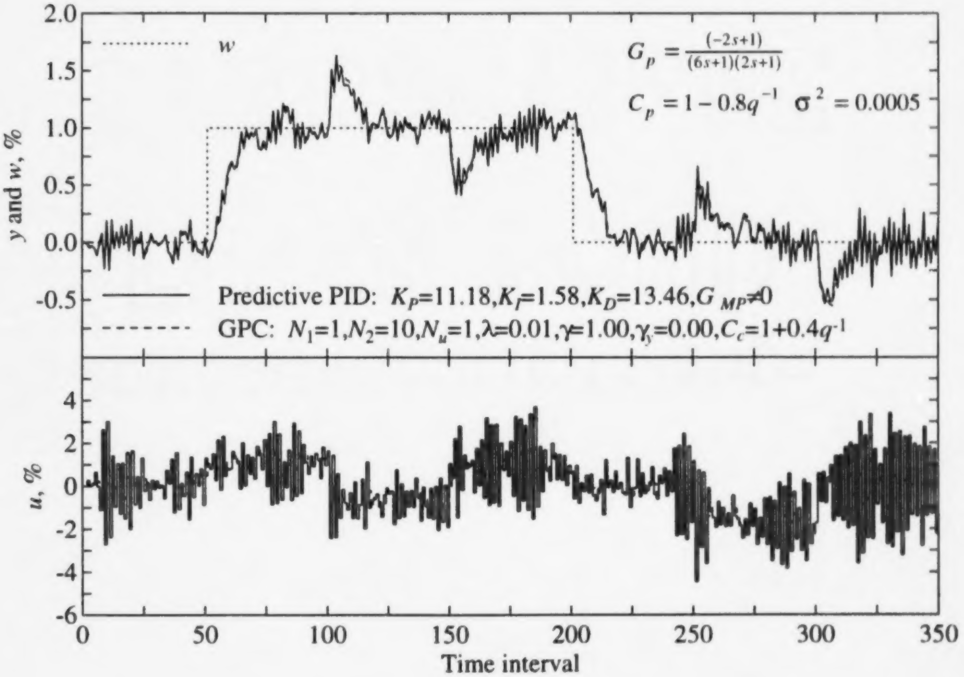


Figure 2.15: Response of ∞GPC and ∞PID with C=1+0.4q^{-1}

2.10 Experimental Evaluation of Predictive PID

Simulation studies are an effective technique to demonstrate controlled behavior for linear plant models. Physical processes consist of many characteristics which cannot be modelled exactly such as high order and time varying dynamics. Since the practical application of control algorithms is for the control of physical processes, it is imperative that a thorough evaluation of controller performance include experimental work. Performance of GPC and γGPC for the control of pilot scale processes is well documented (Clarke, 1991; McIntosh et al., 1991; Kwok and Shah, 1994). A thorough experimental evaluation of predictive PID to show the effect of different controller constants would merely duplicate previous work because predictive PID is equivalent to γGPC for regulatory control. In this section an experimental comparison of GPC, predictive PID and ∞PID is presented.

2.10.1 The Light Bulb Process

Figure 2.16 shows the schematic diagram of the experimental equipment used in this section. The plant consists of a 100 watt light bulb which is powered by a variable 0-120 volt AC source. A J type thermocouple is horizontally mounted approximately one mm above the highest point of the light bulb. The process dynamics are dominated by natural convective heat transfer from the bulb surface to the thermocouple when the fan is not running (which is the nominal case). The heating process of the filament and the bulb surface contribute to the high order dynamics of the overall process. The process is inherently nonlinear because the relationship between voltage and power is nonlinear ($P = V^2/R$). In addition, the prime mode of heat transfer between the filament and the bulb surface is radiation which is highly nonlinear. The simplified radiation heat transfer relationship is given by

$$q_{rad} = A\varepsilon\sigma_{SB}\left(T_{fil}^4 - T_{bulb}^4\right) \tag{2.10.1}$$

where q_{rad} = heat transferred from the filament to the bulb surface, A = filament area, ε = filament emissivity, σ_{SB} = Stefan-Boltzmann constant, T_{fil} = filament temperature and T_{bulb} = bulb surface temperature. Because the dominant component of the overall process is natural convection, air movement in the room has a dramatic effect on the measured temperature. Therefore, the variance of the measured temperature is inconsistent and sometimes very large. A

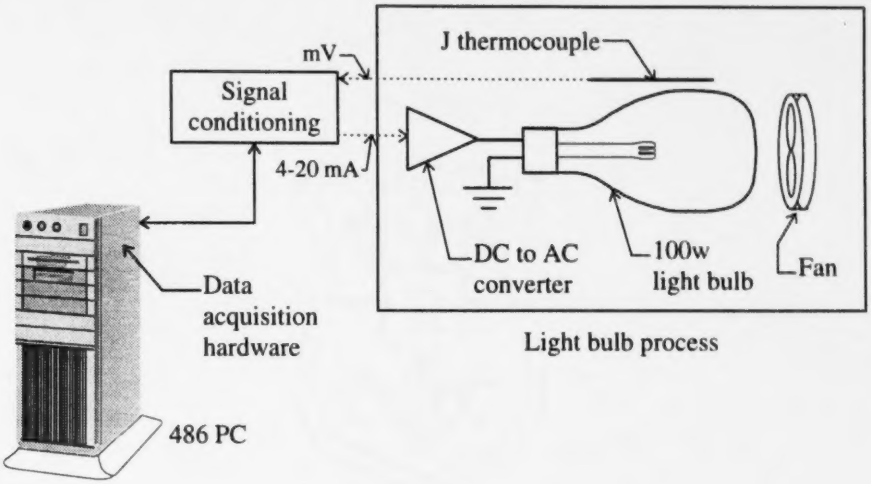


Figure 2.16: Schematic diagram of the light bulb process.

manually controlled fan is used to introduce step disturbances to the temperature.

The data acquisition hardware consists of a PC chassis mounted card which has 12 bit resolution for analog to digital conversion. Thermocouple linearization, signal amplification and transient protection are performed by the signal conditioning hardware as shown in Figure 2.16. A custom program written at the University of Alberta which runs under the LabVIEW® software development system was used for control of the light bulb process. A brief description and an evaluation of LabVIEW® for implementation of advanced control can be found in chapter 6.

2.10.2 Open Loop Analysis

As mentioned in section 2.10.1, the light bulb process dynamics are nonlinear, high order and noisy. Since the purpose of this experiment was not to test a linear controller on a significantly nonlinear process, a linearization table was used to linearize the process. Figure 2.17 shows that the steady state temperature versus controller output relationship is nonlinear with a significant amount of hysterisis. The average of the steady state temperatures at each controller output in Figure 2.17 was used for the linearization table.

A wide range of guidelines for the choice of the sampling interval have been proposed as documented by Seborg *et al.* (1989). A sampling interval that is one tenth of the dominant time constant satisfies most guidelines for digital control and was selected for this work. Since the dominant or first order time constant for the light bulb process was estimated to be 77 seconds, a sampling interval of eight seconds was chosen.

A second order plant model was chosen because the plant is inherently higher order and the maximum allowable order for the predictive PID algorithm is two. The open loop temperature response to step changes in controller output is shown in Figure 2.18. The Matlab® System Identification Toolbox (Ljung, 1992) was used to determine the following ARMAX model of the process.

$$y(t)=\frac{q^{-1}\left(0.0324-0.0048q^{-1}\right)}{1-1.3360q^{-1}+0.3995q^{-2}}u(t-1)$$

$$C=1-0.6109q^{-1}-0.0953q^{-2}$$

The continuous model estimate based on the eight second sampled discrete transfer function is

$$\frac{y(s)}{u(s)}=\frac{0.4352e^{-8s}(4.8s+1)}{(10.1s+1)(65.0s+1)}$$

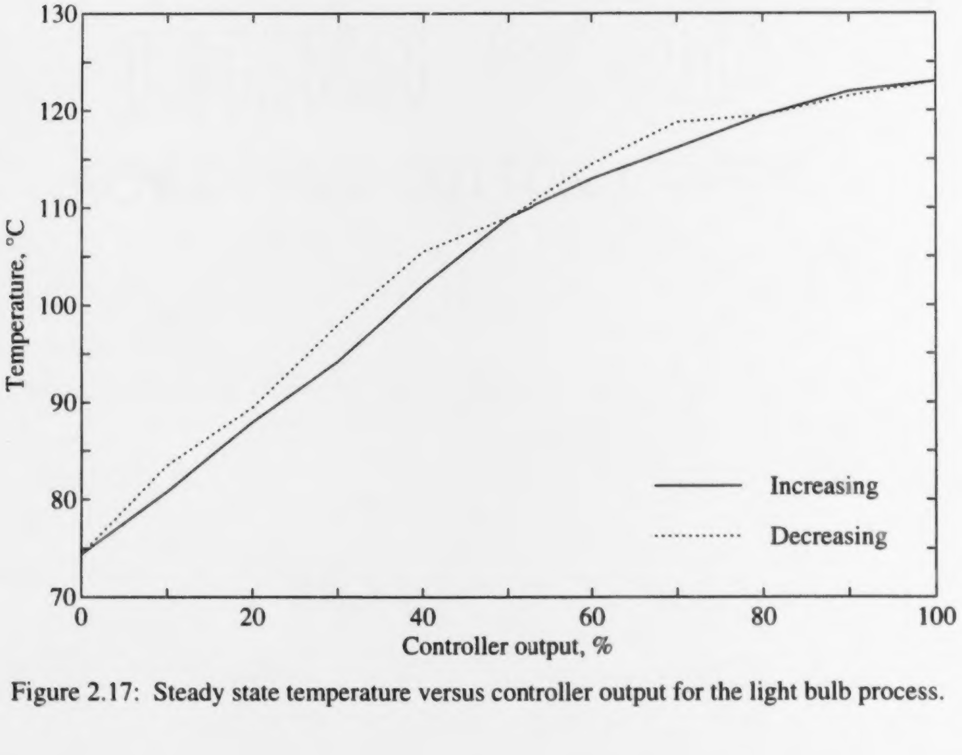


Figure 2.17: Steady state temperature versus controller output for the light bulb process.

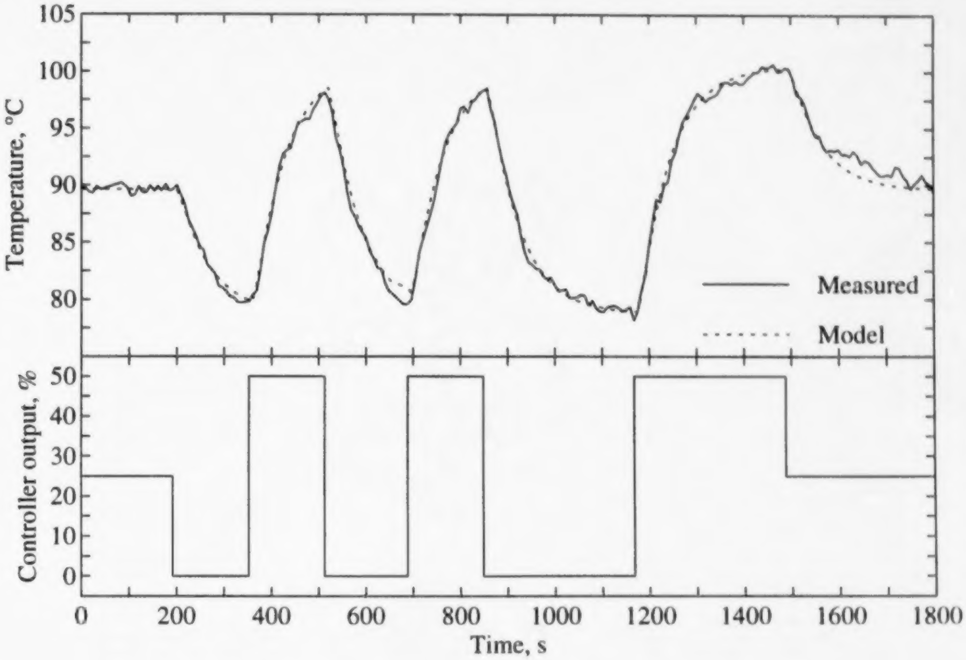


Figure 2.18: Open loop response of the light bulb process.

Although the light bulb process is noisy, the linearized open loop response is a good fit to the above second order ARMAX model as shown in Figure 2.18.

The open loop response (sampled at eight second intervals) to the fan disturbance presented in Figure 2.19 shows that the cooling dynamics of cooling with the fan operating are considerably slower than the dynamics of heating when the fan is stopped. In addition, the variance of the measured temperature is significantly higher when the fan is on as shown by

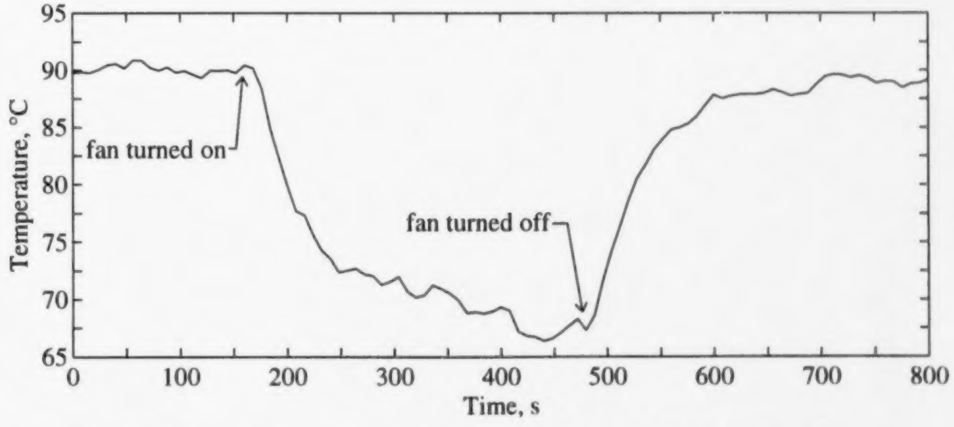


Figure 2.19: Temperature response to the fan disturbance for the light bulb process.

Figure 2.19. This is likely due to swirling air currents over the light bulb surface which is caused by the fan. The duration of the fan disturbance is 320 seconds which is the same duration as used for the closed loop experimental results presented in the following section.

2.10.3 Closed Loop Comparison of GPC and Predictive PID

The objective of the closed loop runs presented in this section is to compare the experimental servo and regulatory performance of GPC, predictive PID and ∞PID in order to corroborate the theoretical and simulation results presented previously. The effect of different GPC and γGPC controller tuning parameters is not studied here as a thorough evaluation already exists (McIntosh *et al.*, 1991; Kwok and Shah, 1994). Because of the inconsistent variability of the light bulb process, an absolute performance measure such as the integral of the square of the error is not used to quantitatively compare each run. The focus is on the relative performance of each controller rather than the absolute performance. A combination of suggested tuning guide lines (McIntosh *et al.*, 1991) and *ad-hoc* measures were used to select the controller constants for GPC and the same controller constants were used for the predictive PID controller. Each run consists of a setpoint change and a fan disturbance which is 320 seconds in duration.

The GPC, predictive PID and ∞PID closed loop controlled response of the light bulb process is shown in Figure 2.20, Figure 2.21 and Figure 2.22, respectively. The controller output of GPC is more aggressive than the controller output of predictive PID following the setpoint changes which results in a slightly faster servo response. This can be explained by the choice of the C_c polynomial which is $1\text{-}0.75q^{-1}$. The final value of C_c is 0.25 which effectively detunes the servo performance of the predictive PID controller. This behavior is consistent with the simulation results presented in section 2.7. From a practical perspective, the setpoint changes for GPC and predictive PID are very acceptable because there is no overshoot and the settling time is 100 seconds or less which is just slightly longer than the open loop time constant of 77 seconds. The ∞PID servo response is significantly slower than both GPC and predictive PID as shown by the results displayed in Figure 2.22. The C_c polynomial was chosen to be $1\text{-}0.7q^{-1}$ for the ∞PID controller which explains the sluggish response. This ∞PID control loop may be a good candidate for including proportional action on setpoint changes as discussed in section 2.7.

Rejection of the fan disturbance for the GPC and the predictive PID controllers is very similar as shown by the responses in Figure 2.19 and Figure 2.20. Small differences in the temperature responses can be attributed to the high variability while the fan is running. The

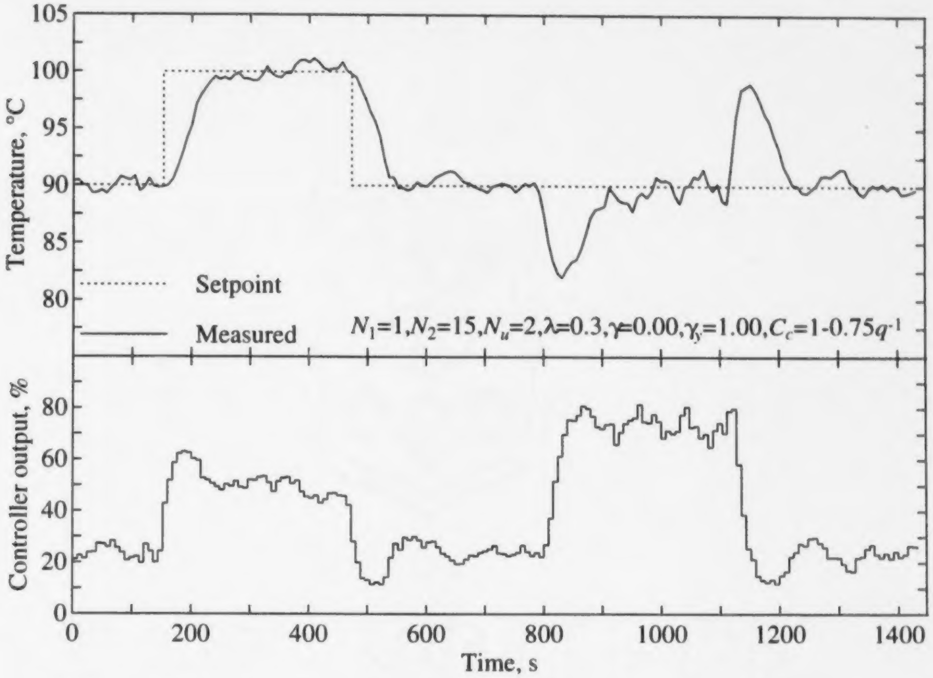


Figure 2.20: GPC servo and regulatory response to the light bulb process.

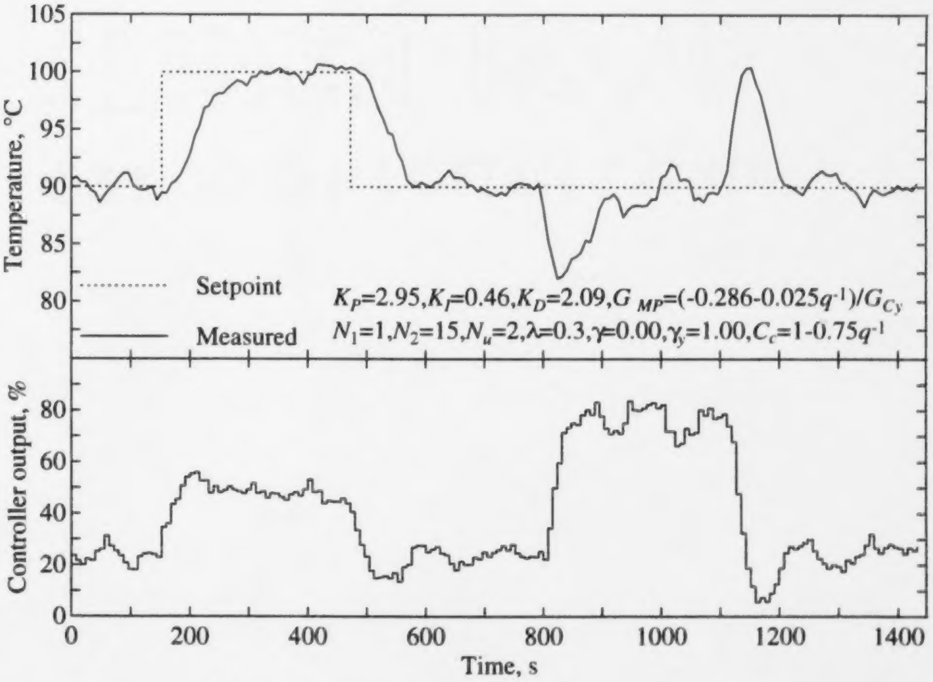


Figure 2.21: Predictive PID servo and regulatory response to the light bulb process.

similar regulatory response of GPC and predictive PID is consistent with the theoretical analysis in section 2.7. The magnitude of the temperature disturbance is higher when the fan is stopped compared to when the fan is operating which agrees with the previous observation of the fan disturbance dynamics in section2.10.2. Considering that the operation of the fan causes the temperature to fall below 65 °C under open loop operation, the regulatory performance of GPC and predictive PID is excellent. Although the regulatory performance of ∞PID is slightly inferior to that of GPC and predictive PID, it is still very acceptable as shown in Figure 2.22. The maximum temperature deviation is about 2 °C higher than the corresponding GPC response during the fan disturbance. Considering the simplicity of the ∞PID control algorithm, the experimental performance is very good.

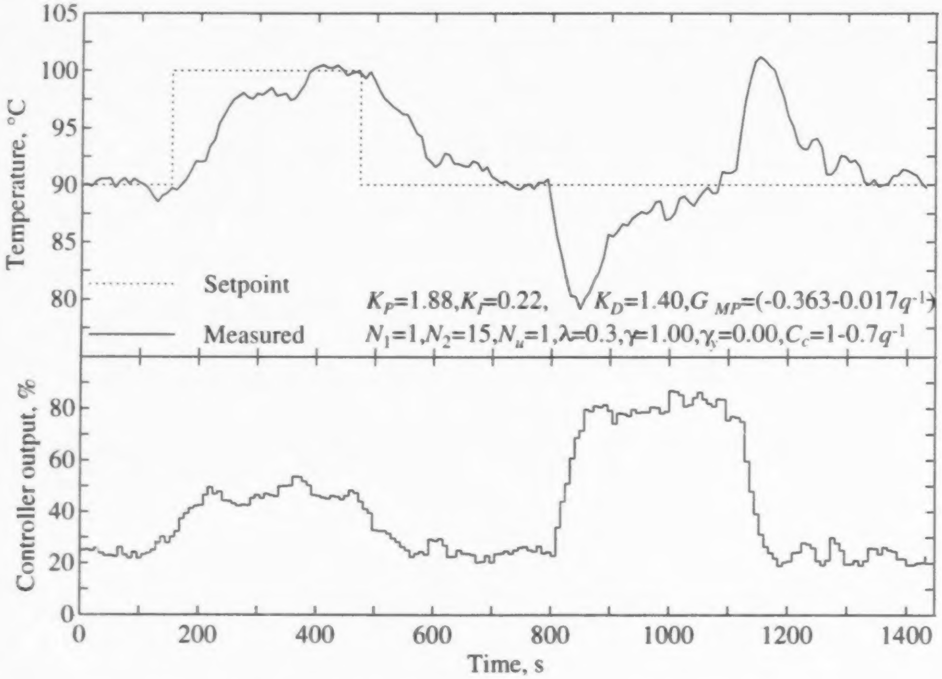


Figure 2.22: ∞PID servo and regulatory response to the light bulb process.

2.11 Industrial Application of Predictive PID

The ultimate evaluation of controller performance is its acceptance in an industrial setting. Industrial corporations are primarily motivated by profit margins and competition in global markets which means that the plant must operate near the highest possible efficiency.

Acceptance of a control scheme upgrade by industry therefore implies superior performance compared to the previous scheme. As mentioned previously, predictive controllers such as DMC are rapidly gaining popularity as a replacement of classical PID control. However, one of the major hurdles of the employment of DMC is that specialized or custom software must be purchased and implemented either as a stand alone application or a complex user defined function in the existing control computer. The proposed predictive PID control scheme in this chapter is intended to allow the implementation of advanced control in existing control computers without the use of specialized software and hardware.

The application of predictive PID to a key process in an Edmonton area fertilizer plant is presented in this section. A comparison of the performance of the existing PID controller and the predictive PID controller is shown for a step disturbance to the process.

2.11.1 Process Description

The main function of the industrial process described in this section is the production of NO_2 which is used by subsequent processes to produce urea. Some of the operating data are normalized and process identification tags are changed because of the proprietary nature of the process. Figure 2.23 shows the simplified schematic and instrument diagram of the process.

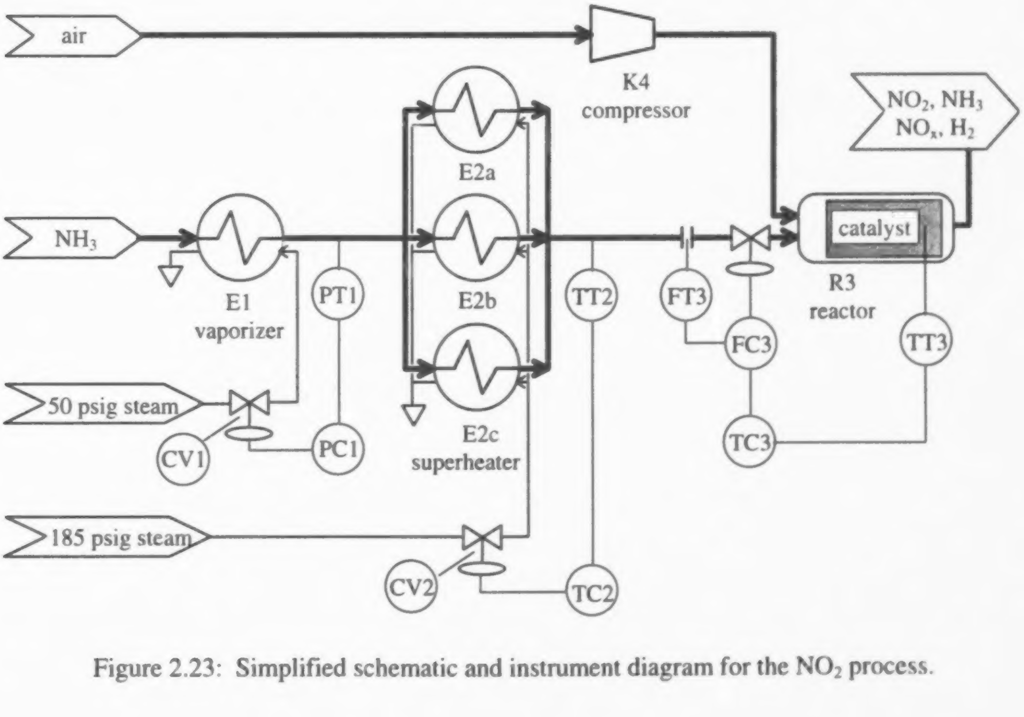


Figure 2.23: Simplified schematic and instrument diagram for the NO_2 process.

Subcooled liquid ammonia enters E1 where it is heated to a saturated vapour by 50 psig process steam. A PID controller is used to regulate the pressure of the saturated ammonia. The saturated ammonia enters E2a, E2b and E2c and is heated to about 300 °F by 185 psig process steam. The existing superheated ammonia temperature controller, (TC2) a conventional PID controller, has been replaced by a predictive PID control law. Superheated ammonia and compressed air enter the highly exothermic reactor, R3, which produces NO_2 and NO_x in a yield of less than 50 %. TC3 is a DMC controller which is cascaded to the ammonia flow controller, FC3. All control loops are implemented on a Honeywell TDC2000 control computer.

2.11.2 The NO_2 Process Problem

The undesired byproducts NO_x (other than NO_2) are produced in significant quantities if the catalyst temperature of R3 drops below 1495 °F (the nominal catalyst temperature is normalized to 1500 °F) while the catalyst degrades prematurely if the temperature exceeds 1505 °F. The result is a very narrow operating range for the catalyst temperature. Control of the catalyst temperature is further complicated by disturbances to the ammonia temperature and the ambient air temperature. The heat exchangers E1 and E2 use process steam which has a significantly higher variance in pressure than utility steam. Although the pressure of the steam header is regulated, there are numerous disturbances caused by load changes in adjacent processes. Furthermore, the temperature transmitter, TT2, is located several metres downstream of the E2 parallel configuration which results in a significant time delay for the TC2 control loop. Interactions between R3 and E2 result in additional disturbances to the ammonia temperature. For example, a one psig decrease in the pressure of the steam in the 185 psig process steam header will cause the superheated ammonia temperature to drop which causes the temperature of the catalyst in R3 to drop. The dynamics of R3 are much faster than E2 which means that the flow rate of ammonia will be quickly increased by TC3 to maintain the catalyst temperature. The increased ammonia flow rate then causes the temperature of the superheated ammonia to fall even more because of reduced retention time in E2. The existing PID controller is tuned to strike a compromise between fast disturbance rejection and good regulation during nominal conditions. As a result, the PID controller oscillates for long periods following disturbances to the steam pressure while the controller output variance is somewhat excessive. In addition, setpoint changes to the ammonia temperature were often observed to cause unstable behavior.

2.11.3 Control Strategy and Implementation

The E2 controller, TC2, was chosen to be upgraded to the predictive PID controller presented earlier because the existing PID controller was known to cause increased variability in the catalyst temperature. In addition, the long time delay and the stochastic nature of E2 are not well suited to conventional PID control. Open loop data and the Matlab® System Identification Toolbox (Ljung, 1992) were used to identify the plant model of E2. The PID controller constants and internal model, G_{MP}, were then computed using equations (2.6.4) and (2.7.12), respectively. The predictive PID algorithm was implemented in a conventional TDC2000 PID control loop with the addition of a simple user defined function for G_{MP} and the removal of proportional and derivative terms from setpoint changes (a built in option). The block diagram of this implementation of predictive PID is illustrated in Figure 2.5. The total time required for this implementation was about one hour (not including modelling of E2).

2.11.4 Comparison of PID and Predictive PID

A comparison of PID and predictive PID control of E2 is shown in Figure 2.24 for a series of step disturbances to the steam pressure. The maximum deviation of the ammonia temperature is approximately the same for both controllers following a five psig step disturbance to the steam pressure. The predictive PID controller rejected both steam pressure disturbances without oscillation in the ammonia temperature. Although the PID controller also rejected the steam pressure disturbances, the ammonia temperature oscillated for several hours following the first disturbance and for over one hour following the second disturbance. After 375 minutes, the controller comparison test was stopped and the loop was placed on manual control because the plant operators were concerned about poor R3 product quality using the existing PID controller. The controller output variance for the predictive PID controller was significantly lower than the PID controller as indicated by the bottom plot of Figure 2.24. Further improvements in regulatory performance for the predictive scheme are likely by increasing controller aggression because the controller output variance is very low. Although increasing controller aggression was suggested by the author, the plant personnel denied the request because the performance of the predictive PID controller was already adequate. Several months after predictive PID was implemented for control of E2, it remains as the controller of choice with a high service factor.

A 15 hour comparison of the control performance using predictive PID and the existing PID during nominal operating conditions is shown in Figure 2.25. The variance of the ammonia temperature is significantly higher when E2 is controlled by the PID controller compared to the period that is controlled by the predictive PID controller. The FCOR (for filtering and correlation analysis) method described by Huang *et al.* (1995) was used to assess the performance of PID and predictive PID in Figure 2.25 relative to minimum variance. A FCOR measure of 1.0 indicates minimum variance while 0.0 indicates open loop response. The PID controlled response resulted in a FCOR measure of 0.14 while predictive PID yielded a measure of 0.26 which represents an 86 % improvement relative to minimum variance. Therefore, predictive PID is superior to the existing PID controller for control of E2.

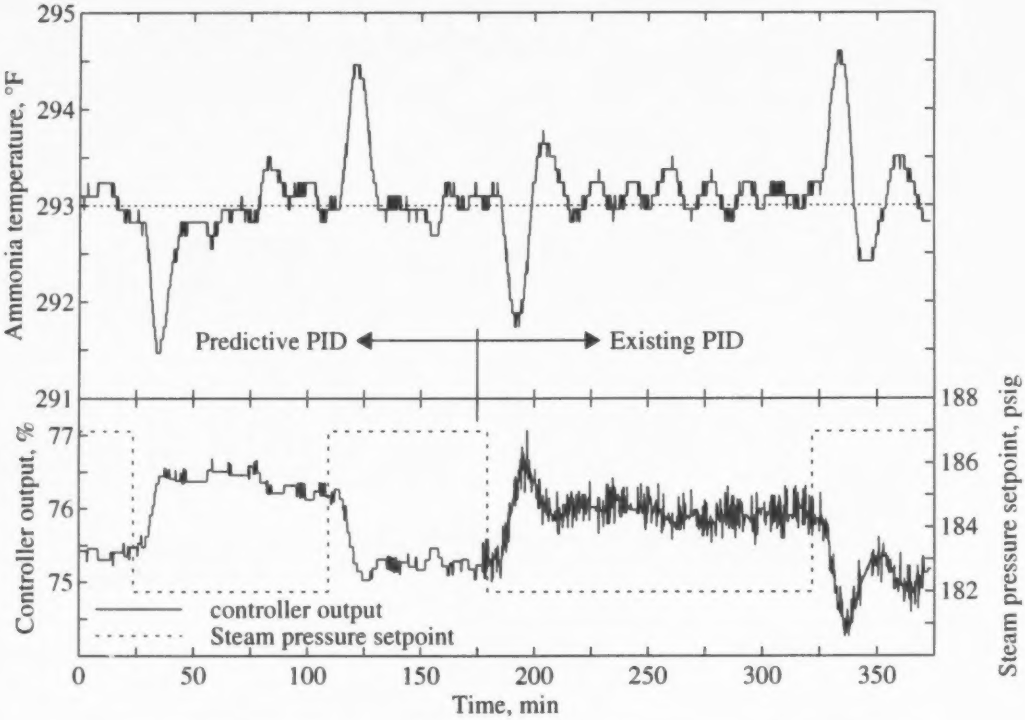


Figure 2.24: Comparison of PID and predictive PID for control of the NO_2 process.

2.12 Conclusions

Long range predictive stochastic PI and PID control laws result from equating the linear polynomials in GPC with PID constants plus an internal model for first and second order plants, respectively. This implies that the proportional and derivative constants in a PID controller are predictive in a long range sense. The internal model G_{MP}, which can be interpreted as a multistep

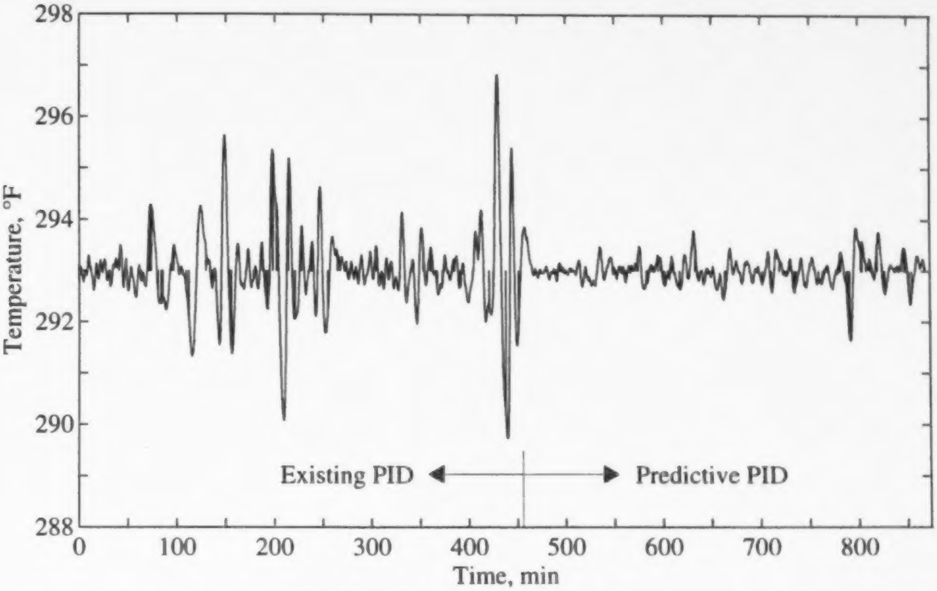


Figure 2.25: Extended regulatory comparison of PID and predictive PID for control of the NO_2 process.

weighted predictor, exists for models with time delay. There are no restrictions on the γGPC tuning parameters N_1, N_2, N_u, λ, γ_y and γ from which the predictive PID controller is based. Predictive PID is equivalent to GPC for all cases except for an approximation of stochastic servo control. In some cases adding the proportional action of predictive PID to setpoint changes results in improved servo response.

- A mean level formulation of PID is determined by setting the finite horizon weight to zero in the predictive PID control law. The resulting ∞PID controller can be expressed as a simple function of ARIMAX model parameters.
- The performance of predictive PID with a finite and an infinite prediction horizon was shown to be excellent in simulation and a pilot scale experimental process.
- Implementation of the predictive PID controller on a TDC2000 control computer for control of a key heat exchanger in a fertilizer plant required only one hour. The performance of the predictive PID scheme was demonstrated to be superior to the existing PID controller for a series of disturbances to the steam pressure and during nominal operation.

References

Åström, K.J., and B. Wittenmark, "On self-tuning regulators," Automatica, Vol. 9, No. 2, pp 185-199, 1973.

Åström, K.J., and B. Wittenmark, *Computer Controlled Systems*, pp 174-196, Prentice-Hall, Englewood Cliffs, NJ, 1984.

Åström, K.J., and T. Hägglund, *Automatic Tuning of PID Controllers*, ISA, NC, 1988.

Banerjee, P. and S.L. Shah, "The role of signal processing methods in the robust design of predictive control," *Automatica,* in press, 1995.

Bitmead, R.R., M. Gevers and V. Wertz, *Adaptive Optimal Control: The Thinking Man's GPC*, Prentice Hall, Englewood Cliffs, NJ, 1990.

Chien, I.L. and P.S. Fruehauf, "Consider IMC tuning to improve controller performance," *Chemical Engineering Progress*, Oct., pp 33-41, 1990.

Clarke, D.W., "Adaptive generalized predictive control," *Proc. Fourth Int. Conf. Chemical Process Control*, pp 395-417, 1991.

Clarke, D.W. and C. Mohtadi, and P.S. Tuffs, "Generalized predictive control - part I. the basic algorithm," *Automatica*, Vol. 23, No. 2, pp 137-148, 1987.

Clarke, D.W. and C. Mohtadi, and P.S. Tuffs, "Generalized predictive control - part II. extensions and interpretations," *Automatica*, Vol. 23, No. 2, pp 137-148, 1987.

Cohen, G.H., and G.A. Coon, "Theoretical consideration of retarded control," *Trans. ASME*, **75**, pp 827-834, 1953.

Cutler, C.R. and S.G. Finlayson, "Multivariable control of a C_3C_4 splitter column," *presented at the national meeting of AIChE*, New Orleans, Louisiana, 1988.

Cutler, C.R., and R.B. Hawkins, "Constrained multivariable control of a hydrocraker reactor," *Proc. American Control Conf.*, Minneapolis, Minnesota, 1987.

Cutler, C.R., and R.B. Hawkins, "Applications of a large predictive multivariable controller to a hydrocraker second stage reactor," *Proc. American Control Conf.*, Atlanta, Georgia, 1988.

Cutler, C.R., and B.L. Ramaker, "Dynamic matrix control -- a computer control algorithm," *Proc. American Control Conf.*, San Francisco, CA, 1980.

Fisher, D.G., "Process control: an overview and personal perspective," *Can. J. Chem. Eng.*, **69,** pp 5-26, 1991.

Gawthrop, P.J., "Self-tuning PID controllers: algorithms and implementations," *IEEE Trans. Automatic Control*, vol. 31, No. 3, pp 201-209, 1986.

Harris, T.J., J.F. MacGregor and J.D. Wright, "An overview of discrete stochastic controllers: generalized PID algorithms with dead-time compensation," *Can. J. Chem. Eng.*, **60**, pp 425-432, 1982.

Henningsen, A., A. Christensen and O. Ravn, "A PID autotuner utilizing GPC and constraint optimization," *Proc. IEEE Conf. Decision and Control*, pp 1475-1480, 1990.

Hokanson, D.A., G. Houk and C.R. Johnston, "DMC control of a complex refrigerated fractionator," *Proc. ISA/89*, Philadelphia, Pennsylvania, October, 1989.

Huang, B., S.L. Shah and K.E. Kwok, "On-line control performance monitoring of MIMO processes," *to be presented at American Control Conf.*, 1995.

Kwok, K., "Long range adaptive predictive control," *Ph.D. Thesis*, University of Alberta, 1992.

Kwok, K., and S.L. Shah, "Long-range predictive control with a terminal matching condition," *Chemical Engineering Science*, vol **49**, No. 9, pp 1287-1300, 1994.

Lambert, E., "The industrial application of long range prediction," *D.Phil Thesis*, Oxford University, 1987.

Ljung, L., *System Identification Toolbox for use with Matlab*, The MathWorks, Inc., Natick, Mass., 1992.

McIntosh, A.R., "Performance and Tuning of Adaptive Generalized Predictive Control," *M.Sc. Thesis*, University of Alberta, 1988.

McIntosh, A.R., S.L. Shah and D.G. Fisher, "Analysis and tuning of adaptive generalized predictive control," *Can. J. Chem. Eng.*, **69**, pp 97-110, 1991.

Park, S., "The application of an optimized DMC multivariable controller to the PCC catalytic cracking unit," *presented at International Symposium of Advanced Process Supervision and Real-Time Knowledge Based Control*, University of Newcastle, Upon Tyne, U.K., 1988.

Richalet, J., A. Rault, J.L. Testud and J. Papon "Model Predictive heuristic Control: Applications to Industrial Processes," *Automatica*, Vol. 14, No. 5, pp 413-428, 1978.

Rivera, D.E., M. Morari and S. Skogestad, "Internal model control. 4 PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, **25**, pp 252-265, 1986.

Saudagar, M.A., D.G. Fisher and S.L. Shah, "Reduced Diophantine formulation for generalized predictive control (GPC)," internal report, Dept. of Chemical Engineering, University of Alberta, 1994.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp, *Process Dynamics and Control*, pp 272-309, John Wiley and Sons, New York, 1989.

Shah, S.L., "Model-based predictive control: theory and implementation issues," presented at ADCHEM '94, Kyoto Research Park, Kyoto, Japan, 1994.

Smith, O.J.M., "Closer control of loops with dead time," *Chemical Engineering Progress*, **53** (5), pp 217-219, 1957.

Tran, D., and C.R. Cutler, "Dynamic matrix control on benzene and toluene towers," *Proc. ISA/89*, Philadelphia, Pennsylvania, October, 1989.

Walgama, K.S., "Multivariable Adaptive Predictive Control for Stochastic System with Time Delays," *M.Sc. Thesis*, University of Alberta, 1986.

Van Hoof, A., C.R. Culter and S.G. Finlayson, "Applications of a constrained multivariable controller to a hydrogen plant," *Proc. American Control Conf.*, Pittsburgh, Pennsylvania, 1989.

Ziegler, J.G., and N.B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, **64**, pp 759-768,1942.

Chapter 3

Control Analysis with Performance Measures

A new overall measure of control performance is proposed in this chapter which combines traditional measures of performance with a new measure of robustness. A graphical example using the overall performance measure is presented to demonstrate the analysis of generalized predictive control (GPC) tuning parameters in the presence of model plant mismatch.

3.1 Introduction

In practice, the overall performance of a controller is based on a combination of many factors including the variance of the controlled variable, variance of the manipulated variable and the ability to minimize the effect of major upsets to the process. Controller constants of model based and PID feedback controllers alike are normally based on some linear model of the process. It is a virtual certainty that the dynamics of the process are somewhat nonlinear and will vary with time therefore robustness to model plant mismatch is essential. The design and subsequent tuning of a controller in industry always involves a tradeoff between robustness and performance (i.e. in order to increase performance, robustness is compromised). Because robustness is not a component of most control objective functions (the H_∞ control law (Francis, 1987) is one exception), *ad-hoc* means are usually employed to determine the controller tuning constants that are relatively intolerant to the expected range of plant dynamics. Although commonly used in industry, *ad-hoc* controller tuning is time consuming and often ineffective. Traditional measures of performance such as integral of the square of the error (ISE) are effective for determining the absolute performance of the controlled variable. However, during periods of model plant mismatch, which often results in low amplitude oscillations in the controlled variable, the ISE criterion may indicate satisfactory or good performance although the loop may only be marginally stable. By combining traditional measures of performance with a measure of robustness for a wide range of controller constants and model plant mismatch, control analysis in simulation will more closely approximate the physical conditions.

The purpose of this chapter is to develop an overall performance measure that consists of the mean square error, mean square incremental control and a margin of robustness based on the small gain theorem. A simple example is also presented to demonstrate the application of the overall performance measure for the analysis of GPC (Clarke *et al.*, 1987) control performance.

3.2 Robustness Measure

There are numerous methods available to analyze the closed loop stability of linear control systems. The purpose of such analysis is either to determine the bounds of the controller constants such that the closed loop stability is guaranteed or to determine if some specific conditions give stable closed loop performance. The norm bounded small gain theorem based on the Nyquist stability theorem is one such technique for the analysis of closed loop stability in the presence of model plant mismatch (Bitmead *et al.*, 1990). Much of the work presented in this section is based on the developments of Banerjee and Shah (1995) in which the small gain theorem is applied for the robustness analysis of GPC.

3.2.1 The Small Gain Theorem

This section gives a brief account of the small gain theorem applied for the robustness of the GPC and PID control laws (see Banerjee and Shah, 1995 for a more thorough treatment). Predictive PID developed in Chapter 2 (also see Miller *et al.*, 1995) has the same closed loop characteristic equation as GPC, therefore the robustness analysis of GPC also applies to the predictive PID control structure.

The PID feedback structure with an additive unmodelled plant perturbation is shown in Figure 3.1 where $\hat{G}_p$ is the plant model and $\tilde{G}_p$ is the model plant mismatch. By grouping the

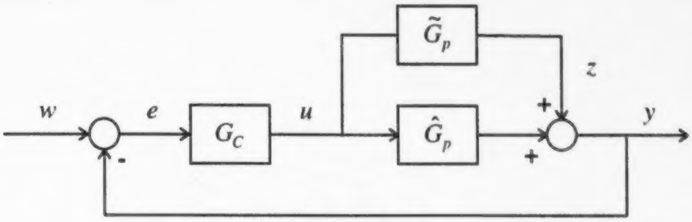


Figure 3.1: Feedback control of a modelled plant with additive perturbation.

modelled feedback control terms with a constant setpoint, Figure 3.1 can be simplified as shown in Figure 3.2 where the PID feedback transfer function is given by the following expression

$$M_{PID} = \frac{u}{z} = \frac{-G_C}{1 + G_C \hat{G}_p} \tag{3.2.1}$$

According to McIntosh *et al.* (1991), the linear formulation of GPC can be expressed as

$$T\Delta u(t) = Rw(t) - Sy(t) \tag{3.2.2}$$

where *T*, *R* and *S* are linear time series polynomials. The GPC feedback transfer function can then be given by

$$M_{GPC} = \frac{-S}{\Delta T + S\hat{G}_p} \tag{3.2.3}$$

In addition, the predictive PID control law described in Chapter 2 (also see Miller *et al.*, 1995) and its corresponding feedback transfer function are given by

$$\Delta u(t) = G_{Cw} w(t) - G_{Cy} y(t) - G_{MP} G_{Cy} \Delta u(t-1) \tag{3.2.4}$$

$$M_{PPID} = \frac{-G_{Cy}}{\Delta\left(1 + G_{MP} G_{Cy}\right) + G_{Cy} \hat{G}_p} \tag{3.2.5}$$

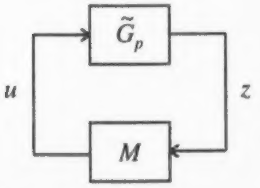


Figure 3.2: Perturbed feedback loop.

With the assumption that the modelled feedback control loop is stable and the additive plant perturbation is stable (i.e. M and $\tilde{G}_p$ are stable), the small gain theorem gives a sufficient condition for closed loop stability which can be expressed as

$$\left\|\tilde{G}_p\left(e^{-j\omega}\right)M\left(e^{-j\omega}\right)\right\|<1 \qquad \forall\omega\in[0,\pi] \tag{3.2.6}$$

or alternatively

$$\left|\tilde{G}_p\left(e^{-j\omega}\right)\right|<\left|\frac{1}{M\left(e^{-j\omega}\right)}\right| \qquad \forall\omega\in[0,\pi] \tag{3.2.7}$$

Banerjee and Shah (1995) have presented a graphical robustness analysis of GPC by plotting the spectrum of the right and left terms of (3.2.7) for various controller constants. Because the small gain theorem is a sufficient but not necessary condition for closed loop stability, the ensuing analysis is potentially conservative.

3.2.2 Robustness Margin

Although the results of the previous section provide the means for a thorough analysis of the stability of linear feedback control systems in the presence of model plant mismatch, the degree or margin of robustness is not quantified. It is therefore the purpose of this section to develop a measure of robustness based on the small gain theorem.

Any violation of (3.2.7) constitutes the potential conditions for unstable closed loop behavior. Thus, a scalar margin of robustness can be given by the minimum difference between the right and left terms of (3.2.7) over the specified frequency range which can be expressed as

$$RM=\begin{cases}\min\left[\left|\dfrac{1}{M\left(e^{-j\omega}\right)}\right|-\left|\tilde{G}_p\left(e^{-j\omega}\right)\right|\right] & \forall\omega\in[0,\pi] \quad \text{if SGT valid}\\ 0 & \qquad\qquad\quad \text{if SGT invalid}\end{cases} \tag{3.2.8}$$

where RM is the robustness margin. If the small gain theorem is violated then there is no concept of a robustness margin hence the reason for setting RM to zero in (3.2.8) for this case. Since the small gain theorem is potentially conservative, the robustness margin is also a conservative measure. A spectral plot for a third order plant and a GPC controller based on a first order model is shown in Figure 3.3 to illustrate the robustness margin. A scalar margin of robustness is a very useful measure because it can be used to determine the margin available for increased controller aggression or model plant mismatch. However, the proposed robustness margin is in terms of a magnitude difference which is a case specific measure.

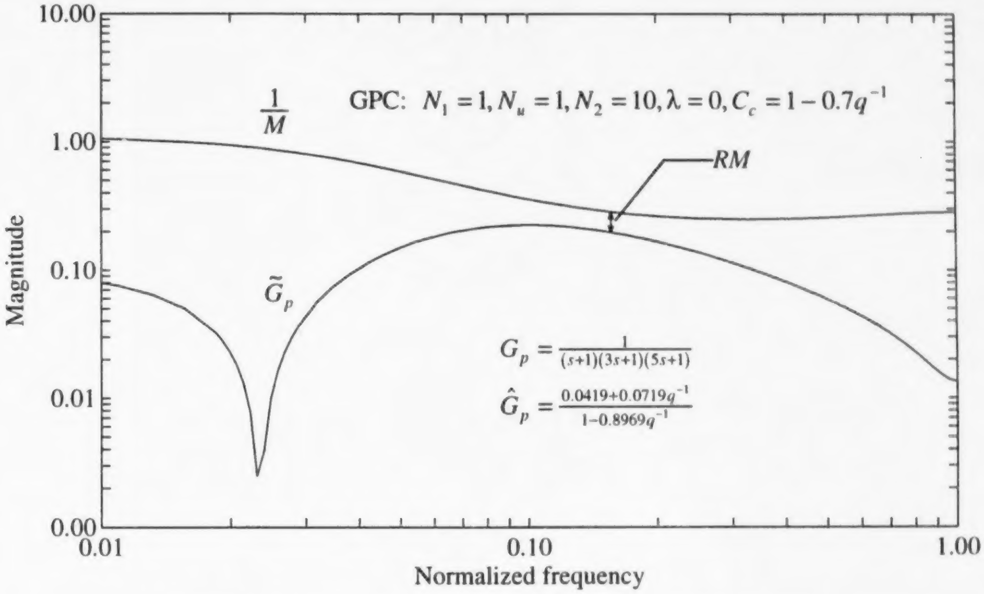


Figure 3.3: Graphical interpretation of the robustness margin.

3.3 Overall Performance Measure

An overall measure of control performance intended for the analysis of closed loop behavior in the presence of model plant mismatch is presented in this section. The most common controller performance measures reported in the literature are ISE, IAE and ITAE which are absolute measures because a value of zero is not practically or mathematically achievable. Depending on the specific objectives of the process, one of these criteria may be preferred over the others. For this study, the ISE criterion is assumed to best reflect the goals of the control system. Although aggressive control action often improves the performance of the controlled variable, it is undesirable in practice because of high variability of the input to the process and increased control valve wear. A penalty on the incremental controller output should be included in the overall performance measure to reflect the desired low controller output variance. In addition, a penalty on the inverse of the robustness margin will reflect that the most important control objective is stable closed loop behavior. The overall performance measure which includes an absolute performance measure subject to penalties on the plant input variance and the closed loop stability is given by

$$P = \frac{1}{N}\sum_{t=1}^{N}\left[w(t) - y(t)\right]^2 + \frac{\lambda_p g_s^2}{N}\sum_{t=1}^{N}\left[\Delta u(t)\right]^2 + \psi\,\frac{1}{RM} \tag{3.2.9}$$

where N is the run length, λ_p is the incremental control penalty, g_s is the gain of the plant model and ψ is the penalty on the inverse of the robustness margin. The square of the plant model gain is included in the second term in order to remove the gain dependence of the control penalty (McIntosh, 1991). The incremental control penalty and the robustness penalty are chosen in accordance to the process objectives.

3.4 Examples

Several illustrative examples are presented in this section in order to demonstrate the overall performance measure for the analysis of control behavior. It is not the intent of this section to perform an exhaustive case study or justify the choice of controller constants.

The inverse of (3.2.9) with $\lambda_p = 0.5$ and $\psi = 0.1$ is computed in each example from a simulation of an ARIMAX plant with $\sigma_\xi^2 = 0.0005$, $C = 1\text{-}0.8q^{-1}$ and consistent disturbances and setpoint changes. In practice, the simulation should be designed to closely emulate the conditions of the process under study. The inverse of (3.2.9) was chosen for the surface plot because the resulting surface is more conducive to a visual analysis of closed loop performance. In addition, unstable closed loop behavior is indicated by $1/P = 0$.

3.4.1 Example 1

In the first example, the same plant and model as in Figure 3.3 are used to demonstrate the overall performance relative to the GPC controller constants λ, N_u and a first order C_c polynomial. Figure 3.4 shows the performance surface for $N_u = 1$ as λ and C_c vary. Optimum performance is observed for $C_c = 1\text{-}0.75q^{-1}$ and $\lambda = 0$ although the peak is relatively sharp. As λ is increased the performance measure is lower while the peak at $C_c = 1\text{-}0.75q^{-1}$ also diminishes. When $N_u = 2$, the maximum performance occurs when $C_c = 1$ and $\lambda = 5$ as shown in Figure 3.5. Based on the performance objective (3.2.9), increasing N_u from one to two results in a degradation of closed loop performance which shows that increasing the controller aggression

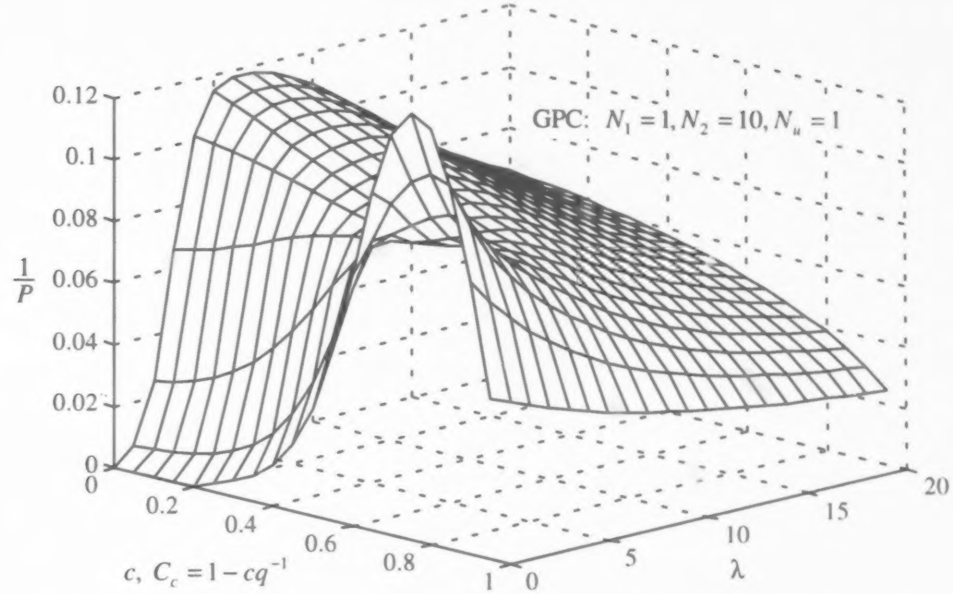


Figure 3.4: Performance surface for GPC control of a third order plant with $N_u = 1$.

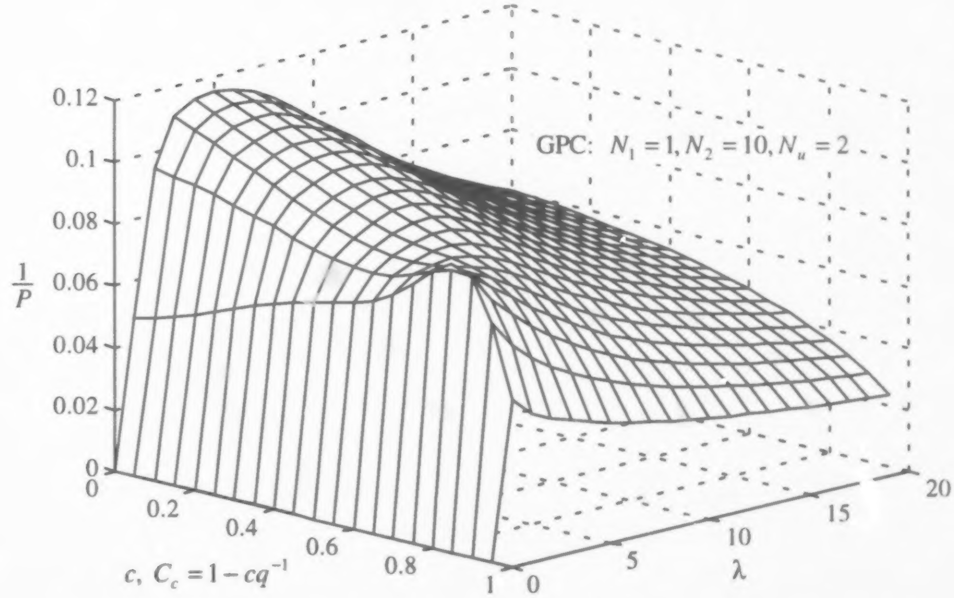


Figure 3.5: Performance surface for GPC control of a third order plant with $N_u = 2$.

does not always give improved performance. Figures 3.4 and 3.5 compress the information from over 800 simulations into two surfaces which gives a deep insight into controller tuning.

3.4.2 Example 2

In this example, the effect of a single GPC controller parameter is shown for varying degrees of time delay mismatch. The first order plant $G_p = \frac{e^{-ds}}{10s+1}$ and model $\hat{G}_p = \frac{e^{-5s}}{10s+1}$ are used while the colored noise, simulation pattern, and performance measure parameters are the same as in Example 1. Figure 3.6 shows the performance surface as the C_c polynomial and the plant time delay varies. At values of c less than 0.88, unstable performance results from even small mismatches in the time delay. This information is crucial for controller tuning if the plant dynamics are known to vary. The performance as λ and the time delay varies is shown by the surface in Figure 3.7. As shown, even small values of λ are capable of stabilizing the closed loop performance for this example although the performance measure is somewhat lower than the use of C_c in Figure 3.6. Larger values of λ significantly detune the controller enough to make the closed loop performance relatively invariant to a time delay mismatch. For this example, it is clear that both C_c and λ weight are required to balance robustness and performance.

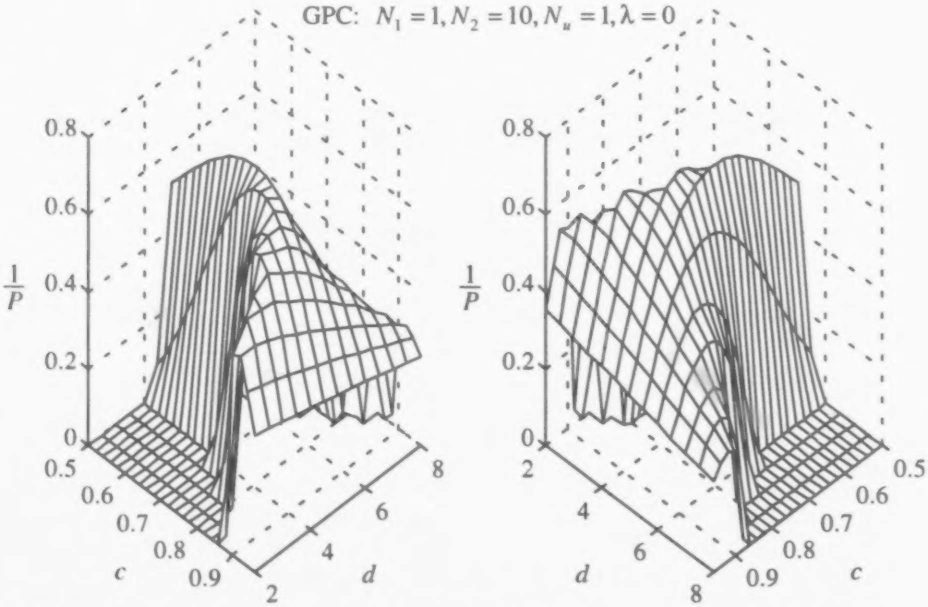


Figure 3.6: Performance surface for GPC control of a first order process as C_c and the time delay varies.

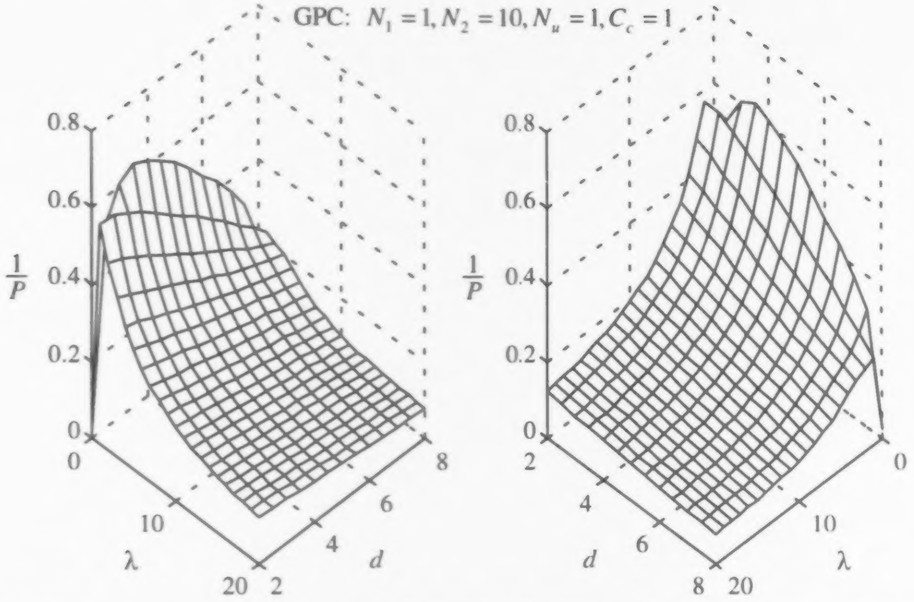


Figure 3.7: Performance surface for GPC control of a first order process as λ and the time delay varies.

3.4.3 Example 3

The final example studies the effect of model plant mismatch on the performance of a fixed parameter GPC controller. The performance surface as a function of time delay and steady state plant gain for the same plant and model as used in Example 2 is displayed in Figure 3.8. A combination of λ and C_c are used to provide stable performance for a ±50% gain mismatch and a ±3 second time delay mismatch. Values of λ and C_c for the GPC controller were chosen based on the results of the analysis in Example 2. It is surprising that the mismatch case, $d = 2$ and $g_s = 1.5$, results in the best performance while the zero mismatch case yields a significantly lower performance. To further investigate the effect of controller tuning on the performance in the presence of model plant mismatch, a single normalized value of the performance in Figure 3.8 (perhaps the sum of each performance value) could be plotted as two controller constants vary. This would essentially allow two more degrees of freedom in the analysis of controller performance although the computing requirements would be very high.

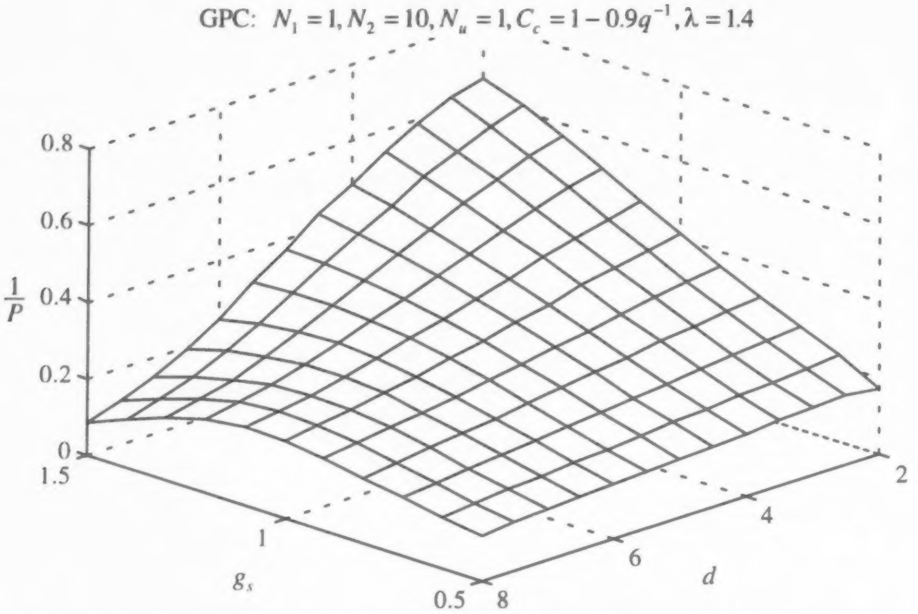


Figure 3.8: Performance surface for GPC control of a first order process with a varying time delay and gain.

3.5 Conclusions

- A new scalar robustness margin based on the small gain theorem is developed which is well suited to analysis of closed loop stability in the presence of model plant mismatch.
- An overall performance measure which combines a control performance measure with penalties on the control output variance and the margin of robustness was presented.
- Computing the overall performance measure as a function of two parameters results in a surface which essentially compresses the information of many simulations into a single plot that is conducive to a visual analysis of control behavior.

References

Banerjee, P. and S.L. Shah, "Estimation of model-plant uncertainty and its role in robust design of predictive control," *to appear in Automatica, May, 1995.*

Bitmead, R.R., M. Gevers and V. Wertz, *Adaptive Optimal Control - The Thinking Man's GPC*, Prentice-Hall, New York, 1990.

Clarke, D.W. and C. Mohtadi, and P.S. Tuffs, "Generalized predictive control - part I. the basic algorithm," *Automatica*, Vol. 23, No. 2, pp 137-148, 1987.

Francis, B.A., *A Course in H_∞ Control Theory*, Springer-Verlag, Berlin, 1987.

McIntosh, A.R., S.L. Shah and D.G. Fisher, "Analysis and tuning of adaptive generalized predictive control," *Can. J. Chem. Eng.*, **69**, pp. 97-110, 1991.

Miller, R.M., K.E. Kwok, S.L. Shah and R.K. Wood, "Development of a stochastic predictive PID controller," *to be presented at the American Control Conf.*, Seattle, WA, 1995.

Chapter 4

Adaptive Predictive PID[1]

A new adaptive predictive PID controller based on the generalized predictive control (GPC) law with a terminal matching condition is presented in this chapter. The PID constants and the internal model are chosen optimally by equating the discrete PID control law with the linear form of GPC. The result is a long range predictive control law with a model based PID structure that can be implemented in any industrial computer control system. First and second order plant models yield, respectively, PI and PID control laws. The plant model order is therefore restricted to a maximum of two although there is no restriction on the choice of GPC tuning parameters. A recursive least squares algorithm based on an upper diagonal factorization method is employed to recursively update the model upon which the predictive PID controller is based. The proposed adaptive predictive PID controller was applied to a steam heated stirred tank heater which exhibits significant time delay. Excellent regulatory response of the proposed scheme is demonstrated for a series of disturbances some of which result in a significant change in plant dynamics.

---

[1] A version of this chapter has been accepted for presentation as: Miller R.M., S.L. Shah and R.K. Wood, "Adaptive Predictive PID," *Proc. ISA/95*, Toronto, April 25-27, 1995

4.1 Introduction

The demand for tighter control of processes with significant non-linearities and time varying dynamics has increasingly motivated research in the field of self tuning and adaptive control over the last twenty years. The two classes of controller adaptation, self tuning or autotuning and adaptive control, correspond to periodic automatic controller tuning and continuous controller adaptation, respectively. The current focus of industrial PID control is in the area of self tuning or autotuning on demand from the operator (Åström and Häggland, 1988). Commercial autotuners including the Foxboro Exact™ and the Satt Control Instruments® autotuner perform on demand autotuning based a set of heuristic rules applied to a perturbation of the process. Such controllers are very useful for the initial commissioning of control loops or when the operator recognizes a change in process dynamics but this class of controllers is not intended to track time varying or non-linear plant dynamics. Moreover, PID control is not well suited to control of plants with non-minimum phase behavior and long time delays.

For an adaptive controller to be effective in industry it must exhibit superior performance to non-adaptive control during periods of time varying plant dynamics. The adaptive controller must also exhibit reliable regulatory performance during periods when the dynamics are time invariant. Adaptive long range predictive control (LRPC) has been shown to be effective for control of processes with non-linearities (McIntosh *et al.*, 1991), time varying dynamics (Clarke, 1991) and multivariable interactions (Shah *et al.*, 1987). The popularity of LRPC is increasing as noted in the literature (Shah, 1994), however, PID remains the work horse of industrial process control (Åström and Häggland, 1988). A LRPC control strategy with a model based PID structure was recently proposed by Miller *et al.* (1995) which was intended to utilize the power of a LRPC control law in existing industrial computer control systems.

The focus of this chapter is to extend the long range predictive PID controller developed in Chapter 2 to adaptive control using a recursive least squares (RLS) technique based on an upper diagonal (*UD*) factorization (Niu *et al.*, 1992). The *UD* decomposition maintains numerical stability of the covariance matrix and provides a wealth of on-line process information. An experimental evaluation of adaptive predictive PID is conducted on a steam heated stirred tank heater which is subject to a series of disturbances. The emphasis is on regulatory control rather than servo control because this is the usual mode of operation in industry.

4.2 The Predictive PID Control Law

This section gives a brief overview of the predictive PID control law. See Chapter 2 or Miller *et al.* (1995) for a thorough treatment of the topic. The predictive PID control law is derived by equating a linear form of generalized predictive control (GPC) (Kwok and Shah, 1994) with a discrete velocity form of PID. Predictions in GPC assume that the process can be adequately represented by a time series model expressed as

$$A\left(q^{-1}\right)y(t)=B\left(q^{-1}\right)u(t-1)+\frac{C\left(q^{-1}\right)}{\Delta}\xi(t) \tag{4.2.1}$$

where A, B and C are polynomials in the backward shift operator (q^{-1}), Δ is an integrator ($1-q^{-1}$) and $\xi(t)$ is a zero mean white noise sequence. The objective function used to establish the GPC control law with steady state weighting (Kwok and Shah, 1994) is composed of a finite horizon error squared term, a control squared term and an infinite horizon error squared term. The GPC control law is formed by minimizing this objective function with respect to future control action. A linear formulation the GPC control law is derived by solving for the current control move as a function of known variables which can be written as

$$\Delta u(t)=Rw(t)-Sy(t)-T_p\Delta u(t-1) \tag{4.2.2}$$

The order of R , S and T_p is nC, nA and nB, respectively where nA, nB and nC correspond to the order of the polynomials in the plant model (4.2.1).

The discrete velocity form of a PID control law, with derivative and proportional action removed from setpoint changes (SP on I), can be expressed as

$$\Delta u(k)=G_{Cw}w(k)-G_{Cy}y(k) \tag{4.3.4}$$

where

$$G_{Cw}=K_IT_s$$
$$G_{Cy}=\left(K_P+K_IT_s+K_D/T_s\right)+\left(-K_P-2K_D/T_s\right)q^{-1}+\left(K_D/T_s\right)q^{-2}$$

where K_P, K_I, K_D and T_s are the non-interacting proportional, integral, derivative and sample interval. For models without time delay and without correlated disturbances (i.e. $C = 1$ and $T_p = 0$), the PID control law (4.3.4) is equivalent to GPC if

$$G_{Cw} = R \qquad G_{Cy} = S \tag{4.2.5}$$

Therefore, a first order plant model results in a predictive PI controller and a second order plant yields a predictive PID controller by equating the linear form of GPC (4.2.2) with the SP on I PID control law (4.3.4). The predictive PI/PID controller can be extended to plants with time delay and correlated disturbances by developing a multistep weighted predictor and a zero order approximation of R. The predictive PID control law is derived by equating (4.2.2) with (4.3.4) for first or second order plant models with time delay and correlated disturbances.

$$\Delta u(t) = G_{Cw} w(t) - G_{Cy} y(t) - G_{MP} G_{Cy} \Delta u(t-1) \tag{4.2.6}$$

where the multistep weighted predictor is

$$G_{MP} = \frac{T_p}{G_{Cy}} \tag{4.2.7}$$

and the zero order approximation to R is

$$G_{Cw} = R(1) \tag{4.2.8}$$

It follows that the model based PID control law (4.2.6) is mathematically equivalent to GPC for regulatory control while servo control is slightly detuned compared to GPC. The PID controller constants are solved for by comparing the coefficients of the relations in (4.2.5). The block diagram of closed loop predictive PID can be expressed as shown in Figure 4.1.

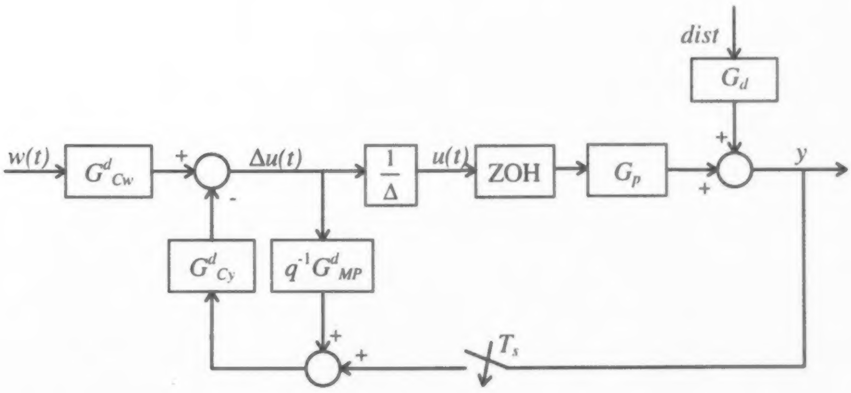


Figure 4.1: Block diagram representation of the predictive PID controller.

4.3 Recursive Plant Model Estimation Using the Augmented Upper Diagonal Identification (AUDI) Method

Least squares parameter estimation schemes have been widely adopted for batch and recursive estimation of linear models from empirical data. These empirical models are used extensively in model based control laws such as GPC. The objective of a least squares estimate of the plant is to minimize the square of the error between the single step ahead prediction and the actual measured value. Recursive least squares (RLS) is the most popular method of estimating time varying process models according to Shah and Cluett (1991). Specific details on the implementation of RLS and its many variants can be found in Shah and Cluett (1991). It is assumed in RLS that disturbances are white (i.e. $C = 1$). Extended least squares (ELS) expands RLS to include estimation of the C polynomial. The disadvantage of ELS is slow parameter convergence due to the iterative nature of updating C. The present work uses a least squares technique because fast convergence is essential to non-linear and time varying plants. In addition, the input-output data can be filtered to remove a bias caused by colored noise.

The main idea behind RLS is to recursively execute the following scheme to update the A and B polynomials of the model described by (4.2.1).

$$\begin{bmatrix}\text{New}\\ \text{Estimate}\end{bmatrix} = \begin{bmatrix}\text{Previous}\\ \text{Estimate}\end{bmatrix} + \begin{bmatrix}\text{Gain}\\ \text{Vector}\end{bmatrix} \times \begin{bmatrix}\text{Prediction}\\ \text{Error}\end{bmatrix} \tag{4.3.1}$$

The gain vector in (4.3.1) is composed of the product of the covariance matrix and the regressor. Recursion of the covariance matrix often results in numerical instability which is well noted in the literature (Bierman 1977, Clarke and Gawthrop 1979, Shah and Cluett 1991). An effective technique to avoid an ill-conditioned covariance matrix is to decompose it into its UDU^T components and update them separately (Bierman, 1977). U and D are the upper diagonal and the diagonal matrices, respectively. The fundamental concept in the augmented UD algorithm (Niu *et al.*, 1992) is the formulation of an augmented regressor given by

$$\phi(t) = \left[-y(t-n) \quad u(t-n-d) \quad \ldots \quad -y(t-1) \quad u(t-1-d) \quad -y(t)\right]^T \tag{4.3.2}$$

where d is the delay and n is the model order. The corresponding parameter vector is

$$\hat{\theta}(t)=\begin{bmatrix} a_n & b_n & \ldots & a_1 & b_1 & 1 \end{bmatrix}^T \tag{4.3.3}$$

The estimated plant model can now be expressed as

$$0=\hat{\theta}(t)\phi(t) \tag{4.3.4}$$

The covariance matrix based on the augmented regressor (4.3.2) is given by

$$C(t)=\left[\sum_{j=1}^{t}\lambda^{t-j}\phi(j)\phi^T(j)\right]^{-1} \tag{4.3.5}$$

where λ is the forgetting factor which is used to place more emphasis on current data or forget old data. The upper diagonal factorization of (4.3.5) yields the following form

$$C(t)=U(t)D(t)U^T(t) \tag{4.3.6}$$

The *U* and *D* matrices contain the model parameters and the loss functions as shown in (4.3.7).

$$U(t)=\begin{bmatrix} 1 & \hat{\alpha}_1^1 & \hat{\theta}_1^1 & \cdots & \hat{\alpha}_1^n & \hat{\theta}_1^n \\ & 1 & \hat{\theta}_2^1 & \cdots & \hat{\alpha}_2^n & \hat{\theta}_2^n \\ & & 1 & \cdots & \hat{\alpha}_3^n & \hat{\theta}_3^n \\ & & & \ddots & \hat{\alpha}_4^n & \hat{\theta}_4^n \\ & & & & \vdots & \vdots \\ & 0 & & & 1 & \hat{\theta}_{2n}^n \\ & & & & & 1 \end{bmatrix} \qquad D^{-1}(t)=\begin{bmatrix} J^0 & & & & & \\ & L^0 & & & 0 & \\ & & J^1 & & & \\ & & & \ddots & & \\ & 0 & & & L^{n-1} & \\ & & & & & J^n \end{bmatrix} \tag{4.3.7}$$

where $\hat{\theta}$ contains the parameter estimates in (4.3.3), $\hat{\alpha}$ contains the parameters of a different (backward) model, *J* is the loss function of the $\hat{\theta}$ model and *L* is the loss function of the $\hat{\alpha}$ model. The superscripts and subscripts in (4.3.7) correspond to the model order and element of $\hat{\theta}$ or $\hat{\alpha}$, respectively. It is clear from (4.3.7) that the augmented *UD* scheme simultaneously estimates the model parameters and loss functions for all model orders from *1* to *n*. Although not employed in this work, a criterion such the Akaike information theoretic criterion (AIC) as described by Ljung (1987) could be used to choose the optimal model order. A thorough explanation of the AUDI structure and its properties can be found in Niu *et al.* (1992). The recursive implementation of the augmented *UD* algorithm with a variable forgetting factor is presented in Table 4.1. An asymptotic data length in the variable forgetting factor as proposed by Niu (1994) is used in order to place more weight on new input-output data that contains more information than the average. Matlab source code for the implementation of AUDI can be found on the Mathworks ftp site: *ftp.mathworks.com*. The *U* and *D* matrices can also be used to provide on-line estimates of the signal to noise (S/N) ratio, noise variance, (Niu and Fisher, 1995), identifiability conditions (Niu and Fisher, 1994) and identification stopping rules (Niu, 1994) with minimal computational effort. Although these on-line estimates are not utilized in this chapter, a practical adaptive control scheme must evaluate the current input output data to establish if sufficient conditions for identification exist. Bad data, strong feedback and severe disturbances may result in insufficient conditions for reliable estimation of plant dynamics. Execution of any identification algorithm in the presence of insufficient conditions could result

Table 4.1: The Recursive AUDI Algorithm with Variable Forgetting.

$U(0)=I_m$ $D(0)=10\cdot I_m$ $\hat{\theta}(0)=[0 \ \cdots \ 0 \ 1]_m^T$	initialization only at startup $m = 2n + 1$ if desired, U and $\hat{\theta}$ can be initialized to the nominal plant values
$\phi(t)=[-y(t-n) \ \ u(t-n-d) \ \ \ldots \ \ -y(t-1) \ \ u(t-1-d) \ \ -y(t)]^T$	regressor
$f=U^T(t-1)\phi(t)$	innovation sequence
$g=D(t-1)f$	
$\lambda(t)=1-\dfrac{[\hat{\theta}^T(t-1)\phi(t)]^4}{N\sigma_\xi^4(1+f^Tg)}$	variable forgetting factor (Niu, 1994) N, K = asymptotic, constant data length $N=K[\hat{\theta}^T(t-1)\phi(t)]^2/\sigma_\xi^2$
$\beta_1=\lambda(t)$	
for j=1 to m	$m = 2n + 1$
$\beta_{j+1}=\beta_j+f_jg_j$	
$D_{jj}(t)=D_{jj}(t-1)\beta_j/\beta_{j+1}/\lambda(t)$	update the D matrix
$\mu_j=-f_j/\beta_j$	
$\nu_{jj}=g_j$	
end	
for j=2 to m	
$\nu_{1:j-1,j}=\nu_{1:j-1,j-1}+U_{1:j-1,j}(t-1)\nu_{jj}$	
$U_{1:j-1,j}(t)=U_{1:j-1,j}(t-1)+\nu_{1:j-1,j-1}\mu_j$	update the U matrix
end	

in unacceptable or unstable performance. A decision on whether to use the current model estimate could be based on a set of heuristic rules which uses the on-line estimates provided within the AUDI framework. However, the development of a set of general practical rules for on-line identification remains a relatively untouched but essential area of research. The AUDI method has tremendous potential in the future development of practical on-line identification of plant dynamics.

4.4 Adaptive Control Strategy

From section 4.2, the control objective of predictive PID is minimization of the square of the prediction errors and control variance over some long range horizon. The adaptive form of the predictive PID controller is established from the general objective of automatic control which can be expressed as minimizing the following criterion with respect to future control action (Shook *et al.*, 1992)

$$J_{AC} = \sum_{j=N_1}^{N_2} \left[w - y(t+j) \right]^2 \tag{4.4.1}$$

where w is the setpoint, N_1, is the minimum horizon and N_2, is the maximum horizon. Since future feedback $y(t+j)$ is unknown, a prediction and prediction error are substituted.

$$J_{AC} = \sum_{j=N_1}^{N_2} \left\{ w - \hat{y}(t+j) - \left[y(t+j) - \hat{y}(t+j) \right] \right\}^2 \tag{4.4.2}$$

Expansion of (4.4.2) yields the following form

$$\begin{aligned} J_{AC} = & \sum_{j=N_1}^{N_2} \left[w - \hat{y}(t+j|t) \right]^2 \\ & + \sum_{j=N_1}^{N_2} \left[y(t+j) - \hat{y}(t+j|t) \right]^2 \\ & -2 \sum_{j=N_1}^{N_2} \left[w - \hat{y}(t+j|t) \right] \left[y(t+j) - \hat{y}(t+j|t) \right] \end{aligned} \tag{4.4.3}$$

The first term in (4.4.3) is the LRPC objective, the second term is the long range identification objective and the third term contains non-linear cross-product terms between control and identification. The objective of least squares algorithms (including AUDI) is to minimize the single step ahead errors which is inconsistent with the long range horizon in (4.4.3). One possible solution is to use a long range predictive identification (LRPI) algorithm (Shook *et al.*, 1992; Kwok and Shah, 1991) and neglect the cross-product terms (i.e. third term of (4.4.3)) but such an approach leads to a complex algorithm and demanding computational resources for its implementation. Another approach is to use a least squares estimator and an *ad-hoc* band pass filtering of the input and output data. Since it was shown by Shook *et al.* (1992) that this approach produces results similar to LRPI this method will be employed in this work.

The approach taken to adaptive control here is to combine predictive PID with AUDI and a band pass filter. The implementation of such a scheme is shown in Figure 4.2.

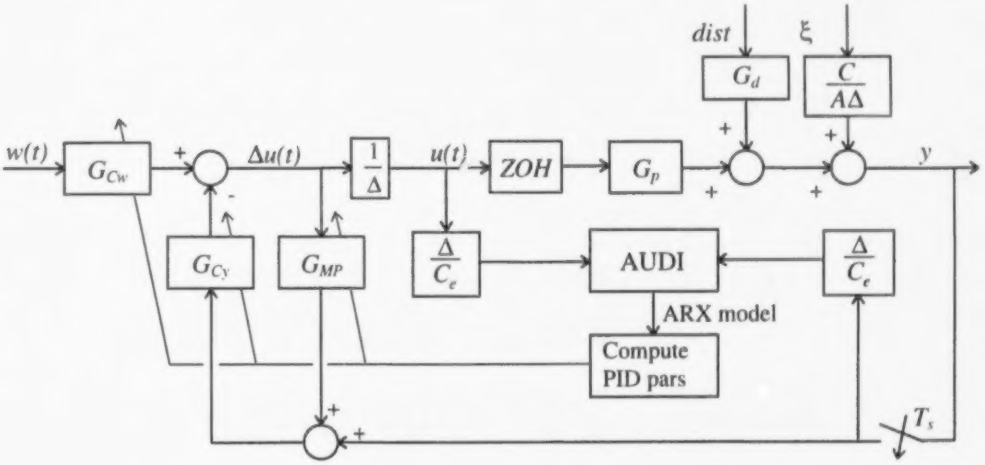


Figure 4.2: Block diagram of adaptive predictive PID in closed loop.

4.5 Experimental Evaluation of Adaptive Predictive PID

The true test of controller performance is an industrial application. Industrial processes are subject to many uncontrolled disturbances and non-linearities which are not present in simulation studies. In addition, the best evaluation of controller performance is in the intended mode of operation which is usually regulation. Non-adaptive predictive PID was shown to be superior to conventional PID for regulatory control of an industrial steam heated heat exchanger in Chapter 2 (also see Miller *et al.*, 1995). Although experimental or pilot scale plants have more control over disturbances than an industrial process, a comparison study of disturbance rejection is better suited to the consistent conditions in a pilot scale plant.

An experimental study of adaptive predictive PID on a pilot scale steam heated stirred tank heater with a significant time delay is presented in this section. The focus of this study is regulatory performance during a series of disturbances. The stirred tank heater is neither noisy nor significantly nonlinear, however, some of the disturbances result in a significant change in process dynamics. This is intended to test the regulatory performance of the adaptive predictive PID controller.

4.5.1 The Stirred Tank Heater Experimental Setup

The schematic diagram of the pilot scale steam heated stirred tank heater used in this study is shown in Figure 4.3. The tank consists of a double-walled glass vessel with a height of 50 cm and an internal diameter of 14.5 cm. Steam is the heating medium which is manipulated by an equal percentage control valve. The process fluid consists of ordinary utility hot and cold water which is mixed by a single impeller in the tank. The temperature transmitter is located about 8.0

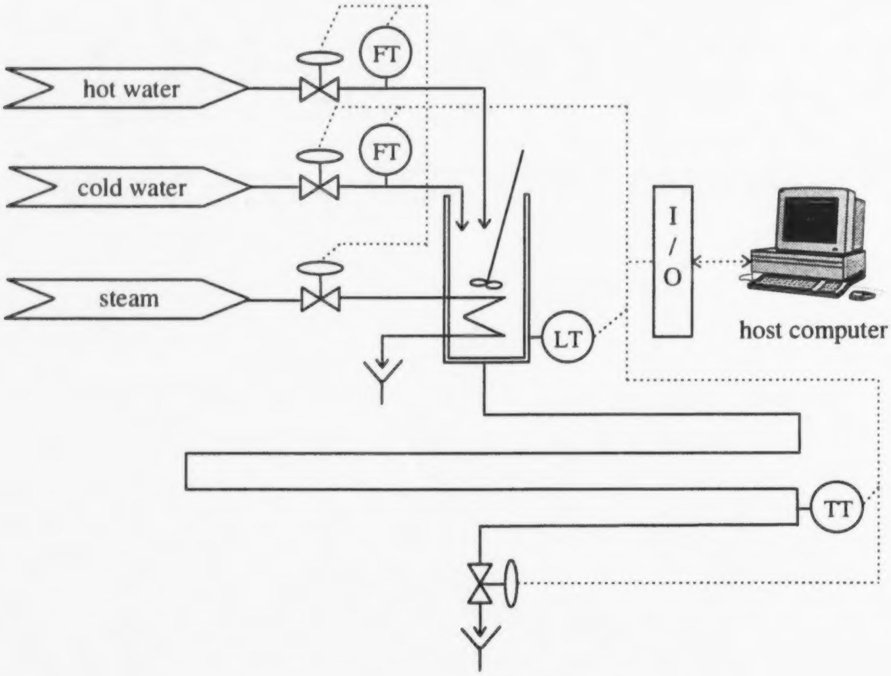


Figure 4.3: Schematic diagram of the stirred tank heater.

metres from the tank exit which results in a time delay of 24 seconds for the nominal case. The I/O subsystem consists of an OPTO 22® Optomux unit which provides an interface between the host computer and the field devices. The host computer consists of a 486 PC running the LabVIEW® software development system (Anon, 1993). All manipulated variables and disturbances on the stirred tank heater were controlled by a custom program written at the University of Alberta running under LabVIEW. Although LabVIEW does not use a robust real-time operating kernel, the stirred tank heater application was observed to be flawless. Communications between the PC and OPTO 22 was provided by an RS-232 serial link.

4.5.2 Control Scheme Implementation

The control configuration consisted of two single input single output loops for control of level and temperature. Water level control was performed by a conventional PID controller which manipulated the cold water control valve signal (control valve signal will be denoted as cvs). The discharge temperature was controlled by predictive PID or adaptive predictive PID which manipulated the steam cvs. Both level control and discharge temperature control are executed by the LabVIEW program. Hot water inlet flow rate, discharge flow rate and level represent the disturbances for this configuration. Table 4.2 shows the steady state operating conditions and temperature transfer function for the nominal case.

Table 4.2: Nominal operating conditions.

Cold water inlet temperature	6 °C
Hot water inlet temperature	45 °C
Tank discharge temperature	35 °C
Level	31 cm
Hot water CV position	8 %
Discharge CV position	100 %

4.5.2.1 Selection of Sampling Rate

There is no universal agreement in the literature on the selection of sampling rate for PID and model based control (Seborg *et al.*, 1989). However, it is agreed that a slow sampling rate with respect to the dominant process time constant can cause aliasing and poor disturbance rejection while a fast sampling rate can cause numerical problems and a high computational load. The dominant open loop time constant and signal-to-noise ratio are two of the major considerations in the selection of sampling rate. Some of the guide lines for sample rate selection are $\frac{1}{6}\tau \le T_s \le \frac{1}{40}\tau$ (Franklin *et al.*, 1990), and $\frac{1}{4}\tau \le T_s \le \frac{1}{10}\tau$ (Isermann, 1981; Stephanopoulos, 1984). The specific considerations for the stirred tank heater are τ = 48s, S/N ratio is high and disturbance rejection is of prime importance. A sampling rate of 4 seconds was chosen as a best compromise to these considerations.

4.5.3 Open Loop Analysis

The open loop responses of the discharge temperature were determined for changes in steam cvs during nominal operating conditions as well as several abnormal steady state conditions. Table 4.3 lists the operating conditions for each open loop discharge temperature response presented in this section. Transfer functions between discharge temperature and steam cvs for the nominal and abnormal operating cases are presented in Table 4.4. The open loop discharge temperature response to step changes in the steam cvs as displayed in Figure 4.4 was used for batch identification (performed by the Matlab System Identification Toolbox (Ljung, 1992)). Although not shown, the discharge temperature response to the same steam cvs input as shown in Figure 4.4 at each abnormal operating case listed in Table 4.3 was used for identification.

Table 4.3: Operating conditions for the open loop stirred tank heater runs.

Figure	Manipulated Variable	Fixed Variables		
4.4	steam*	hot water cvs=8%;	level=31cm;	discharge cvs =100%
4.5	hot water CV	steam cvs =40%;	level=31cm;	discharge cvs =100%
4.6	level	hot water cvs =8%;	steam cvs =40%;	discharge cvs =100%
4.7	discharge CV	hot water cvs =8%;	level=31cm;	steam cvs =40%

* nominal operating conditions

A disturbance in the hot water flow rate and then a subsequent adjustment in cold water flow by the PID controller in order to maintain the level is shown in Figure 4.5. This causes a sustained disturbance in the discharge temperature with a minimal change in plant dynamics as shown by comparison of the nominal and hot water cvs deviation transfer functions in Table 4.4. The small change in dynamics compared to the nominal case is due to a change in the steady state steam requirement which results in a shift along the steam control valve curve.

Table 4.4: Open loop modeling of the stirred tank heater.

Deviation from Nominal Case	Continuous Transfer Function $\left(G_p = \frac{y(s)}{u(s)}, {}^{\circ}C/\%\right)$	Discrete Transfer Function $T_s = 4$ seconds
nominal	$\frac{0.381e^{-24s}}{48.3s+1}$	$\frac{q^{-6}0.0303}{1-0.921q^{-1}}$
hot water cvs=4%	$\frac{0.341e^{-24s}}{41.9s+1}$	$\frac{q^{-6}0.0311}{1-0.909q^{-1}}$
level=22cm	$\frac{0.372e^{-28s}}{40.5s+1}$	$\frac{q^{-7}0.0350}{1-0.906q^{-1}}$
level=40cm	$\frac{0.340e^{-20s}}{47.6s+1}$	$\frac{q^{-5}0.0274}{1-0.919q^{-1}}$
discharge cvs=75%	$\frac{0.503e^{-36s}}{71.8s+1}$	$\frac{q^{-9}0.0273}{1-0.946q^{-1}}$

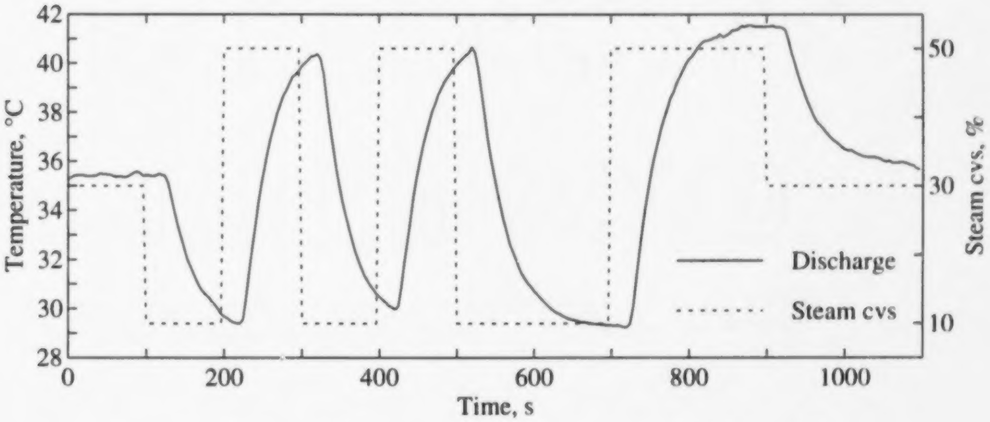


Figure 4.4: Open loop discharge temperature response for the nominal case.

Increasing (decreasing) the level setpoint on the PID controller results in a momentary but significant increase (decrease) in the cold water flow rate to change the water level followed by a slightly higher (lower) steady state cold water flow rate compared to the nominal level case. Therefore, a change in the level setpoint results in an impulse type of disturbance to the discharge temperature which is shown by Figure 4.6. A moderate change in plant dynamics compared to the nominal operating case occurs for a change in level which is shown in Table 4.4. Since the discharge flow rate is a function of liquid level in the tank, the time delay in the transfer function is a function of the tank level. The time delay changes by one sampling interval or four seconds as a result of the level setpoint changes indicated in Table 4.4.

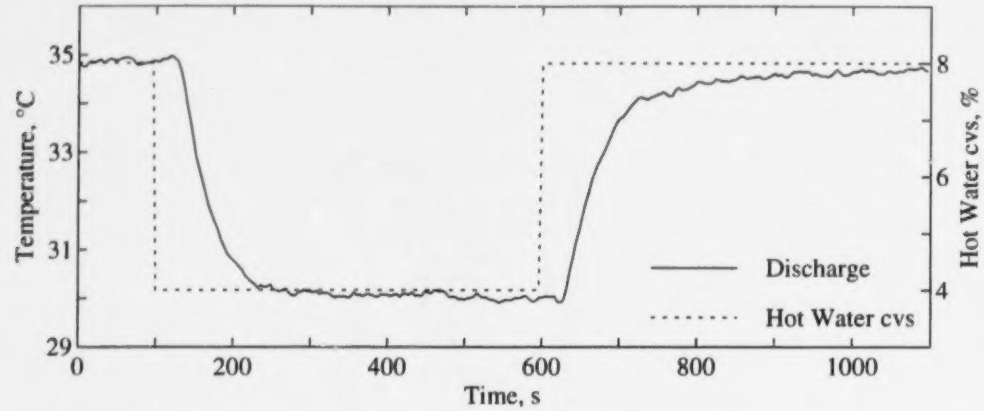


Figure 4.5: Open loop temperature response for the hot water flow rate disturbance.

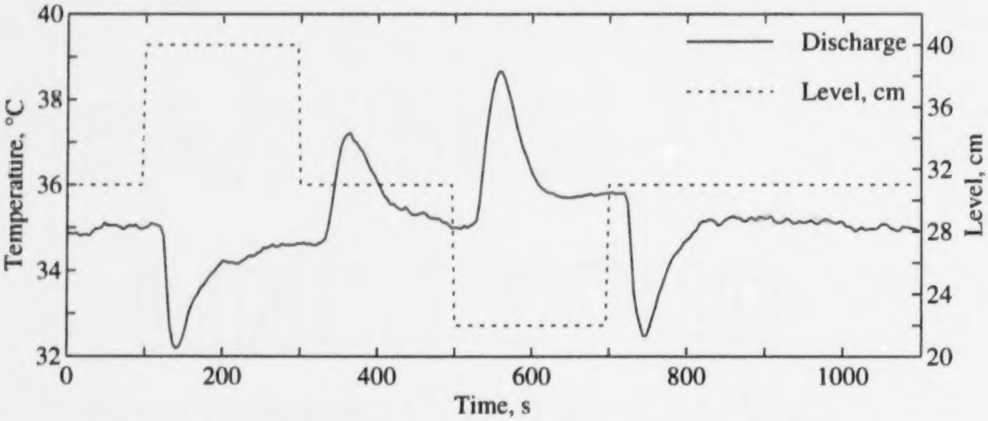


Figure 4.6: Open loop temperature response for the level disturbances.

A step decrease of 25 % in the discharge cvs results in a significant decrease in the discharge flow rate. The PID controller reduces the cold water flow rate in order to maintain the level which results in a severe sustained upset to the discharge temperature as shown in Figure 4.7. The time delay, time constant and plant gain increase by 50 %, 48 % and 32 %, respectively as indicated by the open loop transfer function in Table 4.4. A 50 % increase in the time delay is a severe mismatch compared to the nominal case because the of the phase contribution of the time delay element. The discharge cvs is therefore the most challenging of all three disturbances.

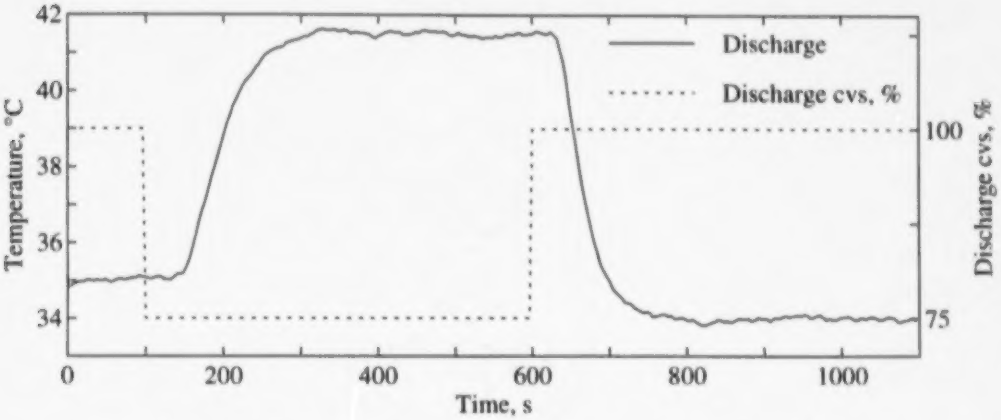


Figure 4.7: Open loop temperature response for the discharge flow disturbance.

4.5.4 Closed Loop Adaptive Predictive PID

The closed loop experimental tests were conducted to study the effectiveness of the non-adaptive and adaptive predictive PID control algorithms for the control of discharge temperature in the presence of input disturbances. For the hot water cvs disturbance, the performance of a conventional PID controller was investigated to provide a benchmark comparison. Constants of the PID controller were established by Cohen and Coon (1953) design relations with *ad-hoc* fine tuning in order to minimize the effect of disturbances on the outlet temperature. The predictive PID algorithms used in all predictive runs were based on the following GPC controller constants: $N_1 = 1$, $N_2 = 30$, $N_u = 1$, $\lambda = 0.8$, $\gamma = 0.3$, $\gamma_y = 1.0$ and $C_c = 1\text{-}0.8q^{-1}$. Readers not familiar with these GPC controller constants should consult Chapter 2 or Miller *et al.* (1995). The C_e filter was chosen to be $(1\text{-}0.4q^{-1})^2$ for all predictive runs. For each test, a setpoint change was first introduced to provide sufficient excitation for the initialization of the covariance matrix. Although this step is obviously not required for non-adaptive algorithms, for consistency, the same format was used for all tests. Although it has been shown in closed loop adaptive GPC that RLS parameter estimates converge approximately to the correct values without *a priori* knowledge (Clarke, 1991; McIntosh *et al.*, 1991; Shook *et al.*, 1992), in practice, on-line identification algorithms are not turned on until the covariance matrix is sufficiently initialized. In view of practical interest and consistency, the ARX parameters in the U matrix were initialized to the nominal values based on the transfer function parameters in Table 4.4 for all adaptive runs. Table 4.5 summarizes the series of closed loop runs in this section.

Table 4.5: Experimental closed loop control of the stirred tank heater.

Figure	Controller Type	Disturbance	Purpose
4.8	Conventional PID	Hot water cvs	sustained disturbance; small Δ in TF
4.9	Predictive PID	Hot water cvs	sustained disturbance; small Δ in TF
4.10	Adaptive predictive PID	Hot water cvs	sustained disturbance; small Δ in TF
4.11	Predictive PID	Level	impulse disturbance; medium Δ in TF
4.12	Adaptive predictive PID	Level	impulse disturbance; medium Δ in TF
4.13	Predictive PID	Discharge cvs	sustained disturbance; large Δ in TF
4.14	Adaptive predictive PID	Discharge cvs	sustained disturbance; large Δ in TF

4.5.4.1 Hot Water Flow Rate Disturbance

As a consequence of the stationary dynamics of a hot water flow rate disturbance, it might be expected that conventional PID control would provide satisfactory control of discharge temperature with the predictive PID controller providing little improvement. However, this is not the case as can be seen in the closed loop responses shown in Figures 4.8 and 4.9.

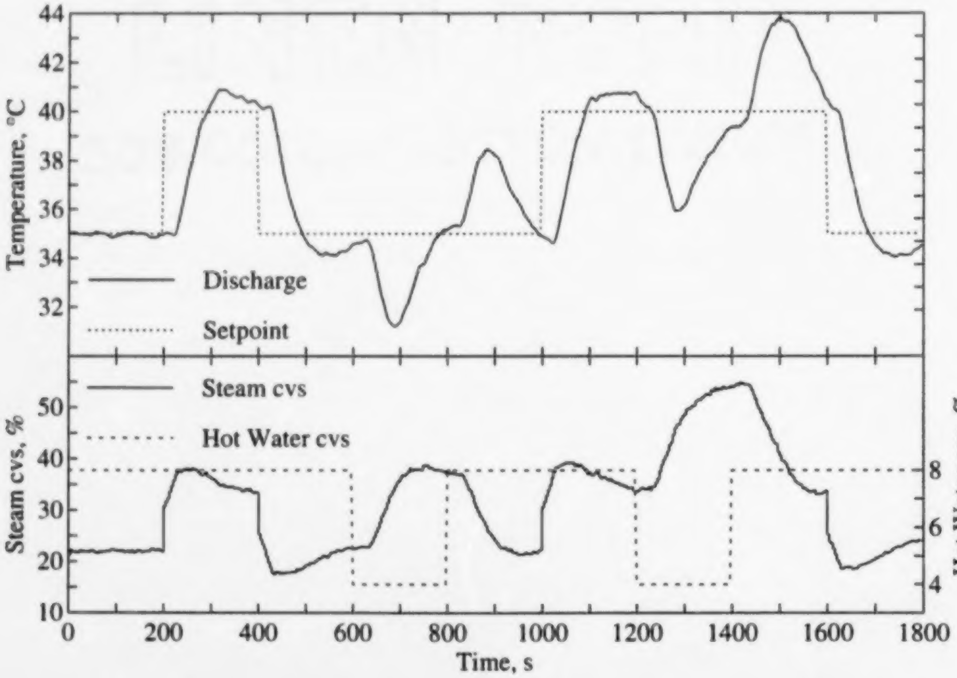


Figure 4.8: PID control of the stirred tank heater with a hot water flow disturbance.

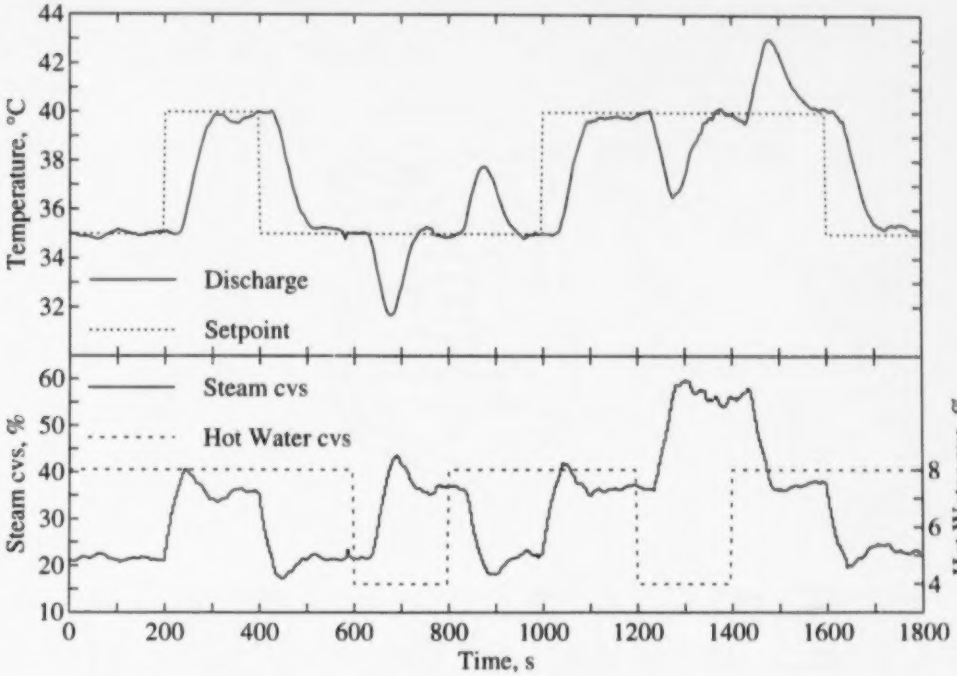


Figure 4.9: Predictive PID control of the stirred tank heater with a hot water flow disturbance.

Examination of the responses in these figures shows that setpoint changes result in an initial overshoot for the PID controlled response compared to the response of the predictive PID controller. In addition, the regulatory performance of predictive PID during hot water flow rate disturbances is clearly superior to conventional PID. Figure 4.10 displays the controlled response for the adaptive form of the predictive PID algorithm. As can be seen, the response of the discharge temperature to the initial setpoint change is inferior to either of the other control algorithms because the covariance matrix is not yet "full enough" of process information for the recursive identification algorithm to provide reliable parameter estimates. Consequently, examination of the performance after 400 seconds of operation shows that the adaptive form provides the best control performance of the three algorithms. The AUDI technique with a band pass filter is therefore able to cope with sustained disturbances that do not result in a major change in dynamics.

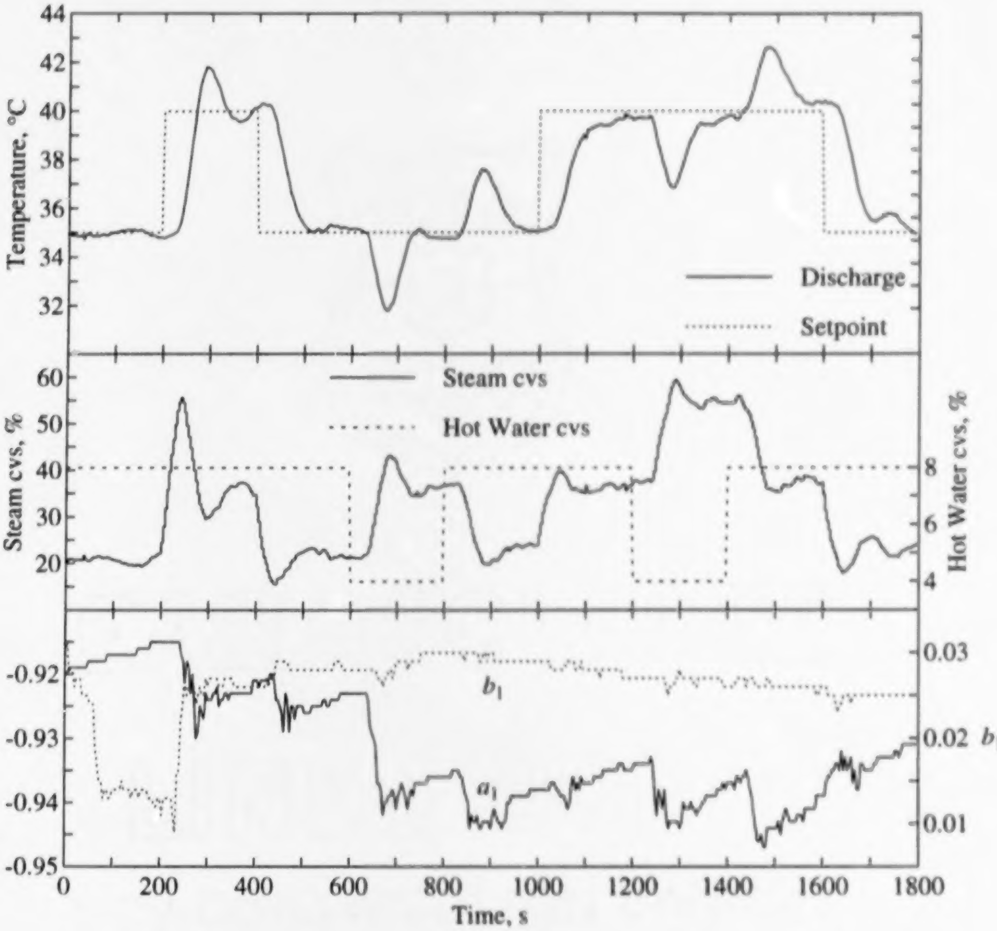


Figure 4.10: Adaptive predictive PID control of the stirred tank heater with a hot water flow disturbance.

4.5.4.2 Level Disturbance

The performance of the predictive PID algorithm for the control of discharge temperature during a level disturbance is displayed in Figure 4.11 and for the adaptive predictive PID algorithm in Figure 4.12. As the water level is decreased, the time delay increases as shown by the transfer functions in Table 4.4. When the level is changed to 22 cm, the predictive PID controller is based on a model that has a shorter time delay than is exhibited in the stirred tank heater. Consequently, it is not surprising that the predictive PID control performance is oscillatory in the 600 to 800 second interval as a result of the reduced phase margin. Although the adaptive predictive PID control scheme is based on a fixed time delay model, recursive adaptation of the

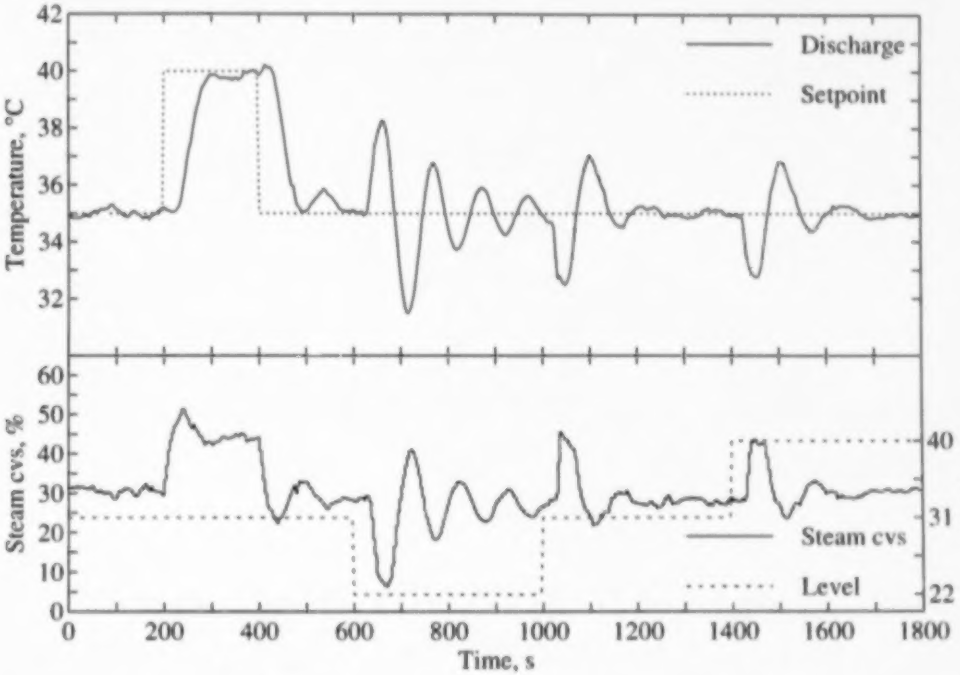


Figure 4.11: Predictive PID control of the stirred tank heater with a level disturbance.

model parameters is able to compensate for the mismatch. Comparison of Figures 4.11 and 4.12 shows superior performance for the adaptive predictive PID controller for each level disturbance. The magnitude and duration of the discharge temperature upsets caused by each of the level changes are significantly lower for the adaptive scheme compared to the non-adaptive controller. It is difficult to compare the parameter trajectories in Figure 4.12 with the level deviation open loop models in Table 4.4 because there is a minor time delay mismatch. However, a distinct shift in parameter estimates following each level setpoint change occurs which indicates that the plant model adapts within the constraints of the fixed time delay structure.

4.5.4.3 Discharge Flow Rate Disturbance

The final disturbance used to test the performance of the predictive PID control algorithms was a change in the discharge flow rate which as shown in Table 4.4 results in the largest change in dynamics of the three different disturbances employed. The control performance that resulted from the predictive PID and adaptive predictive PID algorithms is shown in Figures 4.13 and 4.14, respectively. The discharge temperature response of the fixed model predictive controller is clearly not acceptable during the 600 to 1200 second interval of Figure 4.13. In contrast, the

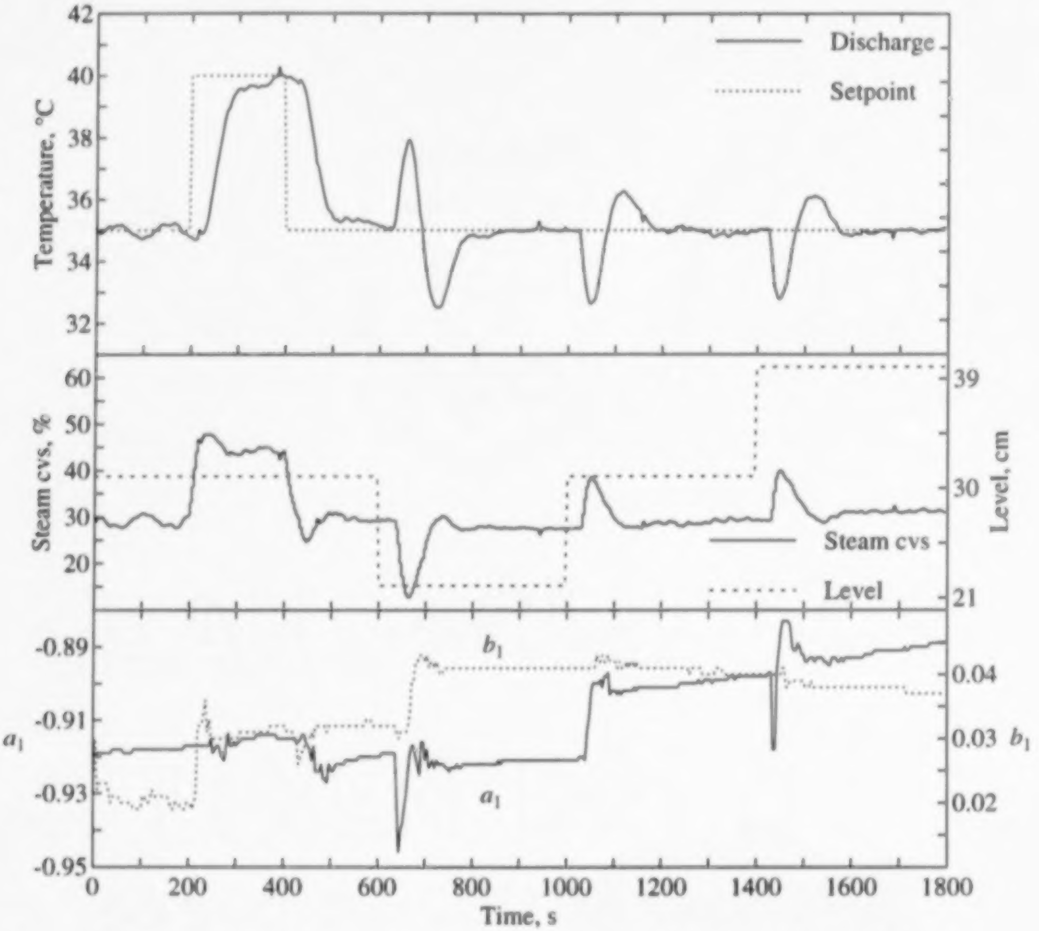


Figure 4.12: Adaptive predictive PID control of the stirred tank heater with a level disturbance.

adaptive predictive PID control algorithm was able to return the discharge temperature to the setpoint after about 450 seconds of the disturbance despite the fact that the time delay of the controller's predictive model remained at 24 seconds. It is obvious from the parameter trajectory plot in Figure 4.14 that in order to accomplish this control behavior, the estimated parameters were adjusted significantly, particularly the a_1 value. The fact that a_1 is adjusted close to its minimum allowable value of -0.995 indicates that the AUDI algorithm is overestimating the time constant in order to compensate for the time delay mismatch. It is also revealing to observe that the gain of the plant model is over estimated by a factor of six which acts to detune the control action. It is this adaptation of the predictive model that eliminates the oscillatory behavior that is evident for the fixed model controller. However, the a_1 coefficient was slow to recover to the nominal model parameter value after the discharge flow is returned to the nominal case which may reduce controller performance if another upset occurred. On-line estimation of time delay in parallel to estimation of the ARX parameters may offer further improvement in performance.

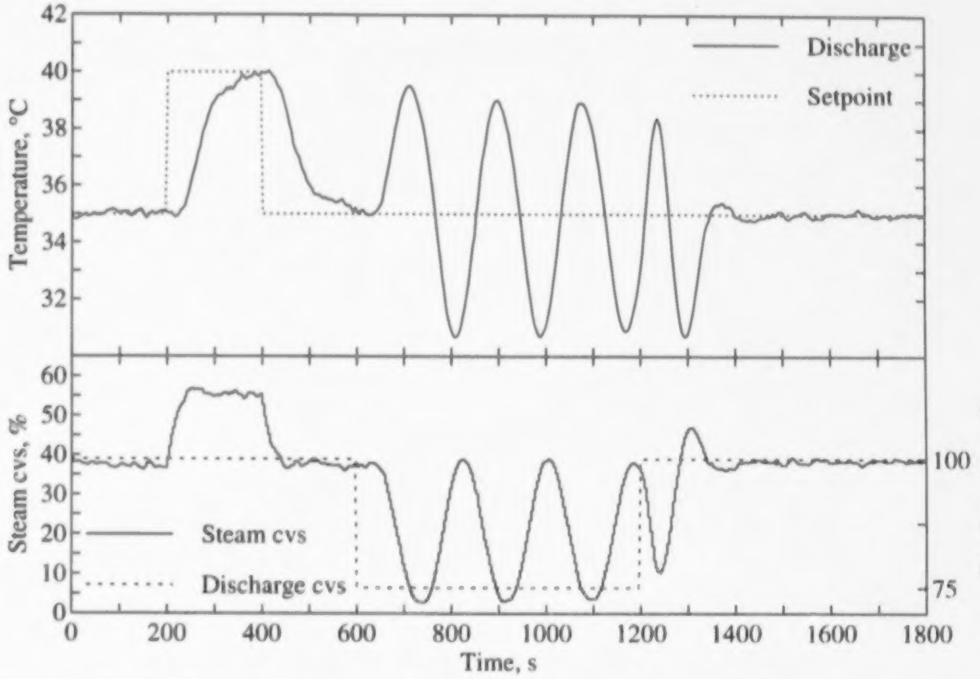


Figure 4.13: Predictive PID control of the stirred tank heater with a discharge flow rate disturbance.

4.6 Conclusions

- The proposed predictive PID control law was shown to exhibit excellent regulatory performance when applied to a steam heated stirred tank heater in the presence of moderate disturbances and moderate model-plant-mismatch.

- A long range predictive PID control law with the augmented *UD* factorization method of on-line model estimation was proposed. An *ad-hoc* band pass filter is suggested to condition the input output data prior to identification to yield an estimated model that is compatible with the long range predictive control law.

- Adaptive predictive control of the stirred tank heater was demonstrated to be excellent for a series of one impulse disturbance and two step disturbances during regulatory control.

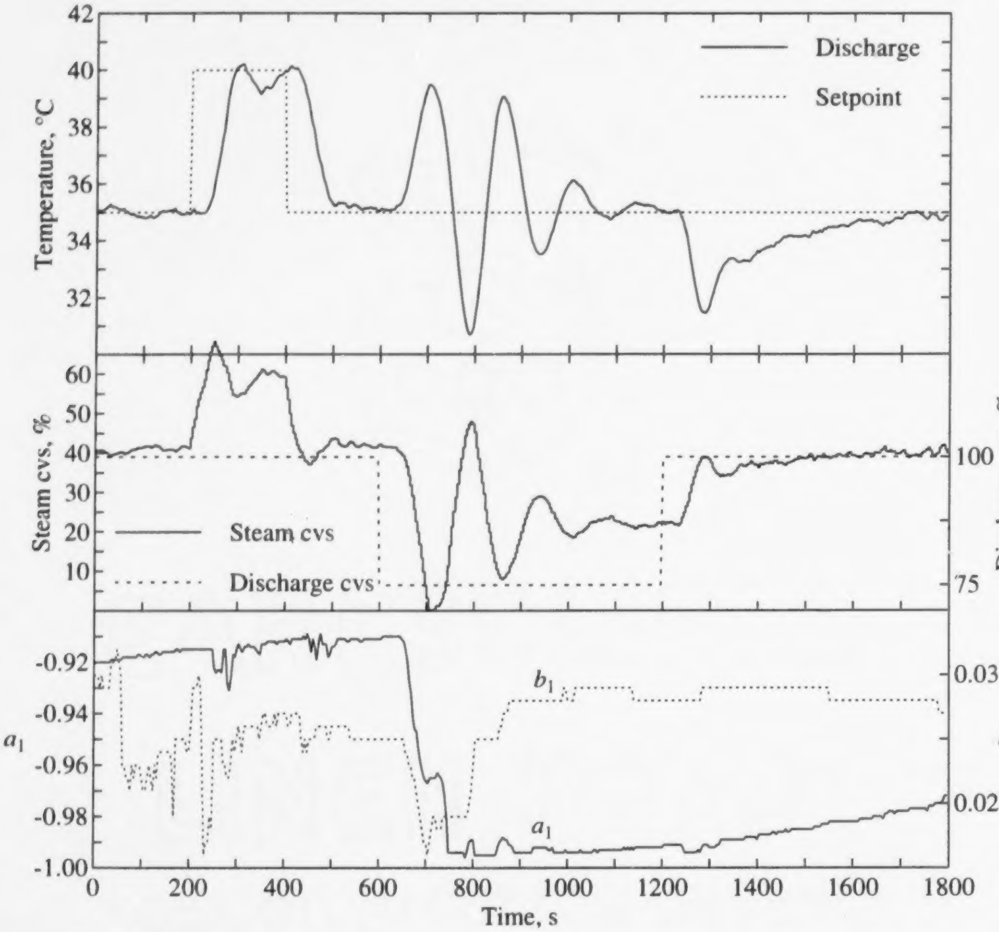


Figure 4.14: Adaptive predictive PID control of the stirred tank heater with a discharge flow rate disturbance.

- The adaptive predictive PID controller based on a model with a fixed time delay showed good experimental results to a process with a significantly varying time delay. On-line estimation of time delay may offer further improvements to the adaptive controller which is the topic of Chapter 5

References

Anon, *LabVIEW for Windows User Manual*, National Instruments Corporation, Austin, TX, 1993.

Åström, K.J., and T. Hägglund, *Automatic Tuning of PID Controllers*, ISA, NC, 1988.

Clarke, D.W., "Adaptive generalized predictive control," *Proc. Fourth Int. Conf. Chemical Process Control*, pp 395-417, 1991.

Cohen, G.H., and G.A. Coon, "Theoretical consideration of retarded control," *Tran. ASME*, **75**, pp 827-834, 1953.

Kwok, K.Y. and S.L. Shah, "Long-range predictive control and identification with steady state weighting," *Proc. American Control Conf.*, pp 2806-2811, 1991.

Kwok, K., and S.L. Shah, "Long-range predictive control with a terminal matching condition," *Chemical Engineering Science*, Vol **49**, No. 9, pp 1287-1300, 1994.

Isermann, M., *Digital Control Systems*, Springer-Verlag, 1981.

Ljung, L., *System Identification Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

Ljung, L., *System Identification Toolbox for use with Matlab*, The MathWorks, Inc., Natick, Mass., 1992.

McIntosh, A.R., S.L. Shah and D.G. Fisher, "Analysis and tuning of adaptive generalized predictive control," *Can. J. Chem. Eng.*, **69**, pp 97-110, 1991.

Miller, R.M., K.E. Kwok, S.L. Shah and R.K. Wood, "Development of a stochastic predictive PID controller," *to be presented at the American Control Conf.*, Seattle, WA, 1995.

Niu, S., D.G. Fisher and D. Xiao, "An augmented UD identification algorithm," *Int. J. Control*, Vol. 56, No. 1, pp 193-211, 1992.

Niu, S. and D.G. Fisher, "Monitoring parameter identifiability during on-line identification," *presented at the 3rd IEEE Conf. Control Applications (also submitted to Int. J. Cont.)*, Glasgow, August, 1994.

Niu, S. and D.G. Fisher, "Simultaneous estimation of process parameters, noise variance and signal-to-noise ratio," *IEEE Trans. Signal Processing*, in press, 1995.

Niu, S., "Augmented UD identification for process control," Ph.D. thesis, University of Alberta, 1994.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp, *Process Dynamics and Control*, pp 272-309, John Wiley and Sons, New York, 1989.

Shah, S.L. and W.R. Cluett, "Recursive least squares based estimation schemes for self-tuning control," *Can. J. Chem. Eng.*, Vol. 69, pp 89-96, 1991.

Shah, S.L., C. Mohtadi and D.W. Clarke, "Multivariable adaptive control without a prior knowledge of the delay matrix," *Systems and Control Letters*, 9, pp 295-306, 1987.

Shah, S.L., "Model-based predictive control: theory and implementation issues," *presented at ADCHEM '94*, Kyoto Research Park, Kyoto, Japan, 1994.

Shook, D.S., C. Mohtadi and S.L. Shah, "A control relevant identification strategy for GPC," *IEEE Trans. Automatic Control*, Vol. 37, No. 7, pp 975-980, 1992.

Stephanopoulos, G., *Chemical Process Control*, Prentice-Hall, NJ, 1984.

Chapter 5

Recursive Time Delay Estimation[1]

A novel on-line time delay estimation technique based on the rationalization of a discrete plant model numerator is presented in this chapter. An auto-regressive with exogenous input (ARX) model with an over parameterized numerator that covers the full expected range of time delay is updated on-line using a recursive least squares technique. The magnitude and sign of the numerator coefficients of the ARX model are used to compute an integer time delay. A low order plant model based on the estimated time delay is recursively updated at each time interval. The proposed time delay estimation technique in combination with a long range predictive control law was applied to a steam heated stirred tank heater with a time varying delay. Experimental results show good time delay tracking during periods of varying dynamics which results in excellent closed loop performance compared to non-adaptive control.

5.1 Introduction

Many chemical and industrial processes exhibit varying time delays which are mainly due to transportation lags and incomplete mixing. The phase contribution of the time delay component can dominate the overall dynamics of the process leading to an unstable closed loop response. The best solution from a control perspective is to reduce the time delay as much as possible by repositioning the measurement transducer closer to the source of the process dynamics. In some cases a reduction in time delay is not possible, for example, a large heat exchanger with a multipass bundle has a significant time delay even if the temperature transmitter is located at the process exit. Also, the cost of moving a measurement sensor may be prohibitive or require a plant shutdown. Therefore, long time delays are part of the process dynamics so the delay must be accepted and dealt with by the control strategy. Model based control laws, such as GPC, are very capable of controlling systems with a time invariant delay (Clarke *et al.*, 1987). However, an unknown variable time delay poses a considerable challenge to any control system even for severely detuned controllers (Dumont *et al.*, 1993). Variable time delays, caused by variable flow rates and mixing among other conditions, occur frequently in industrial processes. For example, black liquor evaporators in the pulp industry are known to exhibit varying time delays due to changing mixing conditions, channeling and dead zones (Dumont *et al.*, 1993). In order for model based control laws to operate effectively, the time delay must be known to some reasonable accuracy. Batch cross-correlation analysis of large quantities of open or closed loop data can provide a good estimate of the average time delay which is adequate for control of fixed delay processes. However, the average time delay estimate is often insufficient for stable control of processes which exhibit a time varying delay. Moreover, selecting the longest estimated time delay of for a process for use with a fixed model controller will result in severely detuned or unpredictable performance. On-line estimation of the process time delay offers a solution to the problem of controlling variable time delayed processes.

---

[1] A version of this chapter has been accepted for presentation as: Miller, R.M., S.L. Shah and R.K. Wood, "Adaptive predictive control employing on-line time delay estimation," *Proc. ISA/95*, New Orleans, 1995.

This chapter begins with a brief review of current recursive time delay estimation techniques for use in adaptive model based control. Following this, a novel time delay estimation technique is proposed which is based on the rationalization of an extended numerator time series model. A comparison study of the proposed technique with variable regression estimation (Elnaggar et al., 1991), via simulation, is presented. Finally, an experimental closed loop evaluation of the proposed time delay estimation method with a long range predictive controller is performed.

5.2 Review of Existing Recursive Time Delay Estimation Techniques

Existing on-line time delay estimation methods generally fall into one of two categories. The first is the development of a plant model that has a flexible structure encompassing the full range of possible time delays (i.e. from d_{min} to d_{max}). The second is the estimation of a distinct integer time delay in terms of sampling intervals for use in a fixed structure model.

5.2.1 Variable Delay Model Structures

In the methods reviewed in this section, the time delay is not specifically estimated but the model structure contains sufficient degrees of freedom to approximate a range of time delay between the process input and the process output. The simplest method of this group is over parameterizing the numerator (known as an extended numerator) of the auto-regressive with exogenous input (ARX) model given by

$$y(t)=\frac{\left(b_1 q^{-d\min}+b_2 q^{-d\min-1}+\cdots+b_{n+(d\max-d\min)} q^{-d\max-n+1}\right)}{1+a_1 q^{-1}+\cdots+a_n q^{-n}}u(t-1)+x(t) \qquad (5.2.1)$$

where $d_{\min}$ and $d_{\max}$ are in terms of sampling intervals. As (5.2.1) is recursively updated, the b coefficients will be weighted such that (5.2.1) approximates the process dynamics including the time delay within the range $d_{\min}$ to $d_{\max}$. Dumont *et al.* (1993) point out that the extended numerator method increases the likelihood of common factors in the numerator and denominator of (5.2.1) which increases the difficulty of the recursive identification problem. Also, the increased number of parameters has several disadvantages. Computational requirements of recursive identification and execution of the control algorithm increase significantly as the number of parameters in the model increase. The requirement for persistent input excitation is higher with a higher order B. In addition, recursion of an extended numerator model converges slowly to time varying dynamics compared to a fixed delay ARX model. Several applications are reported in the literature which are based on an extended numerator model including adaptive control of processes with a varying time delay (Clough and Park, 1985), adaptive control of integrating processes with a varying time delay (Prasad *et al.*, 1985) and adaptive control with a Smith predictor model (Chien *et al.*, 1984).

A popular group of methods involves a low order rational approximation of the time delay. The simplest type in this group is the Padé approximation (Robinson and Soudack, 1970; Gabay and Merhav, 1976; and De Souza et al., 1988) in which the quotient of two low order polynomials are chosen to approximate the Taylor series expansion of e^{-ds}. Other rational approximations to the time delay include Walsh functions (Rao and Siuakumar, 1970), an all-poles approximation (Gawthrop and Nihtila, 1985) a Laguerre series model (Zervos and Dumont,

1988) and a Markov-Laguerre model (Banerjee and Shah, 1995). Dumont *et al.* (1993) reports several industrial applications of adaptive control based on a Laguerre series model of the plant. However, rational approximations to plant dynamics have several disadvantages when used in adaptive control. A distinct delay cannot be identified from the discrete form of the rational function which means the control law must be based on the same rational function. Most rational approximation models are non-minimum phase even for minimum phase processes which may cause difficulties in closed loop control. In addition, the order of the rational function must be proportional to the time delay in order to preserve numerical accuracy of the discretized representation which restricts the application to processes with a short range of time delay.

5.2.2 Distinct Time Delay Estimation

This group of methods recursively estimates a distinct time delay from plant input-output data for use in any model type which has the advantage of not requiring the control law to be redefined in terms of a model with many degrees of freedom. An obvious brute force method to estimate the time delay involves recursive identification of an ARX model for each time delay in the expected range of plant time delays. Therefore, $d_{max} - d_{min} + 1$ models are updated and the optimal choice of the time delay with a resolution of the sampling interval corresponds to the ARX model with the lowest loss function. An advantage of this method is the simultaneous estimation of model parameters and time delay but a major disadvantage is the intensive computational requirements of recursively updating numerous models. In order to reduce the computational load, only a few ARX models could be updated that still cover the entire range of time delays but do not include each interval within the range. Pearson and Wuu (1984) suggest using a spline function to determine the time delay from this reduced set of ARX models although the model parameters that correspond to the estimated delay must still be estimated. Kurtz and Goedecke (1981) propose a time delay estimation method consisting of comparing some function of a fixed delay ARX model with the same function of an extended numerator ARX model. Although this procedure is somewhat complicated, reported simulation results show good estimation of time varying delays in the presence of persistent servo excitation. A method proposed by Ferretti *et al.* (1991) uses the fact that time delay uncertainty can be compensated for by high frequency real negative zeros. Inspection of the phase contribution of these zeros is used to recursively update the time delay estimate in a discrete time ARX model. The authors claim that this method is particularly suited to systems that use a fast sampling rate (< 1 second).

5.2.2.1 Variable Regression Estimation of Process Time Delays

A recursive cross-correlation method, denoted as variable regression estimation (VRE), is suggested by Elnaggar *et al.* (1991) for on-line estimation of the process time delay. The VRE method, described below, uses a first order ARX model,

$$\hat{y}(t) = a_m y(t-1) + q^{-d} b_m u(t-1) \tag{5.2.2}$$

that is a basis for estimation of the prediction error given by

$$\varepsilon^2(t) = [y(t) - \hat{y}(t)]^2 \tag{5.2.3}$$

substituting the model (5.2.2) into the prediction error (5.2.3) yields

$$\varepsilon^2(t) = [y(t) - a_m y(t-1) - b_m u(t-d-1)]^2 \tag{5.2.4}$$

which can be expanded into the following expression

$$\begin{aligned} \varepsilon^2(t) = {} & y^2(t) + a_m^2 y^2(t-1) + b_m^2 u^2(t-d-1) - 2a_m y(t) y(t-1) \\ & -2b_m y(t) u(t-d-1) + 2a_m b_m y(t-1) u(t-d-1) \end{aligned} \tag{5.2.5}$$

The minimum value of the sum of the squares of the prediction errors at the t^{th} sampling instant is given by the expected value of (5.2.5) which is

$$J = E\left(\varepsilon^2(t)\right) = E_0 - 2b_m E_1 \tag{5.2.6}$$

where E_0 and E_1 represent the expected value of the auto-correlation and cross-correlation terms in (5.2.5), respectively.

$$E_0 = E\left[y^2(t) + a_m^2 y^2(t-1) + b_m^2 u^2(t-d-1) - 2a_m y(t) y(t-1)\right] \tag{5.2.7}$$

$$E_1 = E\left[y(t) u(t-d-1) - a_m y(t-1) u(t-d-1)\right] \tag{5.2.8}$$

It is obvious from (5.2.6) that in order to minimize J, E_1 must be maximized which is a strong function of the time delay. Furthermore, Elnaggar *et al.* (1991) indicate that it is advantageous to fix the value of a_m in (5.2.8) to 1.0 for the purposes of time delay estimation. E_1 can now be rewritten as a cross-correlation between incremental plant output and plant input expressed as

$$E_1(t,d_i) = \Delta y(t) u(t - d_i - 1) \tag{5.2.9}$$

Estimation of the time delay from (5.2.9) simply requires computing E_1 for each d_i within the finite range

$$\mathbf{d} = [d_{\min} \quad d_{\min+1} \quad \cdots \quad d_i \quad \cdots \quad d_{\max}] \tag{5.2.10}$$

where $d_{\min}$ is the smallest expected process time delay and $d_{\max}$ is the largest expected delay. As the sampling time goes to zero, the time delay that maximizes E_1 is equivalent to the time delay estimated by extending the tangent to the maximum slope of the step response. The recursive implementation of VRE is illustrated in Table 5.1.

Table 5.1: The recursive VRE algorithm with data forgetting.

$E_1(0,d_i) = 0 \quad \forall d_i$	initialization at $t = 0$
for $i = d_{\min}$ to $d_{\max}$	
$E_1(t,i) = \lambda E_1(t-1,i) + u(t-i-1)[y(t) - y(t-1)]$	compute E_1 for each d_i
end	
$\hat{d}(t) = d_i \wedge E(t,d_i) = \max(\mathbf{E}_1(t))$	$\hat{d}(t)$ maximizes $E_1(t,d_i)$

The delay estimate from VRE can now be used in any recursive identification method to estimate a model of the plant. Elnaggar *et al.* (1991) indicate that the term *variable* in VRE rises from the fact that the structure of the plant model will vary as the time delay estimate changes. Since VRE is based on a first order ARX model, the time delay estimate will be optimal for a first order approximation of the plant dynamics. The VRE time delay estimate will obviously overestimate the time delay best suited to higher order plant models. This fact should not be a problem because the cases in which on-line time delay estimation are important are when the time delay is a dominant factor in the plant dynamics and a first order approximation to the plant should suffice. Computational requirements for the implementation of VRE in Table 5.1 are minimal because only simple multiplications are required. Elnaggar *et al.* (1991) cite several simulation and experimental applications of VRE in combination with a pole placement controller for control of varying time delay processes in the presence of persistent servo excitation. The ability of VRE to track time varying delays is good for square wave servo input although this is a completely unrealistic test. Frequent setpoint changes result in a very high level of plant input excitation which is ideal for identification. However, the main function of process control is regulation of process variables not to significantly magnify the variability of the process. A more realistic evaluation of VRE in closed loop simulation is presented in section 5.4. During regulatory control, plant input excitation is generally very low while disturbances add further adversity to the identification problem. Practical implementation issues such as filtering the estimates, stopping rules and validation are not addressed in the VRE technique.

5.3 Extended Numerator Rationalization for On-line Estimation of Time Delay

In this section, a new on-line time delay estimation technique is proposed for use in closed loop adaptive control. This method, like VRE, estimates an integer time delay in terms of sampling intervals for use with the plant model. The purpose behind the development of yet another time delay estimation technique is to address more realistic problems, such as on-line closed loop identifiability during regulatory control due to low excitation and bad data. The proposed method involves the rationalization of an extended ARX numerator (which will be denoted as ENR) model that is updated by a recursive least squares (RLS) technique based on an augmented upper diagonal factorization method (AUDI) by Niu *et al.* (1992) (also see Chapter 4).

5.3.1 Development of ENR

A first order ARX model with an over-parameterized numerator that contains adequate parameters to cover the variation of the plant delay can be expressed as

$$y(t)=\frac{\left(b_1 q^{-d\min}+b_2 q^{-d\min-1}+\cdots+b_{1+(d\max-d\min)} q^{-d\max}\right)}{1+a_1 q^{-1}}u(t-1)+x(t) \tag{5.3.1}$$

In response to a change in the plant time delay, the AUDI algorithm adapts the coefficients of (5.3.1) such that converged model approximates the new dynamics of the plant. The b_i coefficients will converge to different relative values to reflect the change in plant time delay. Although a_1 initially shifts in response to a plant time delay change, the converged value of a_1 will be close to its nominal value providing that Σb_i converges to its nominal value. Clearly, the relative weights of b_i contain the information required to estimate the plant time delay. Since the value of a_1 is unrelated to the plant time delay and (5.3.1) is used only for time delay estimation,

a_1 in (5.3.1) is fixed to the nominal estimated value rather than recursively estimated. This new model can be expressed as

$$\begin{aligned} y(t)+\bar{a}_1 y(t-1) &= \left(b_1 q^{-d\min} + b_2 q^{-d\min-1} + \cdots + b_{1+(d\max-d\min)} q^{-d\max}\right) u(t-1) \\ &= B_d u(t-1) \end{aligned} \tag{5.3.2}$$

where $\bar{a}_1$ is the nominal value of a_1. Recursion of the AUDI algorithm based on (5.3.2) rather than (5.3.1) has the advantage of reducing the computational requirements slightly because one fewer parameter is updated. In addition, the b_i coefficients of (5.3.2) are forced to adapt more vigorously by fixing the denominator which increases the sensitivity of the time delay information contained in the relative b_i weights. This is illustrated in Figure 5.1 for white input excitation of the plant, $G_p = \frac{q^{-d}0.2}{1-0.8q^{-1}}$, which is subject to changes in the time delay. As expected, the numerator coefficient of (5.3.1) and (5.3.2) which corresponds the current plant time delay converges to 0.2. In addition, the a_1 parameter of (5.3.1) responds to an increase (decrease) in the plant time delay by a sharp decrease (increase) in value with the exception of the first plant

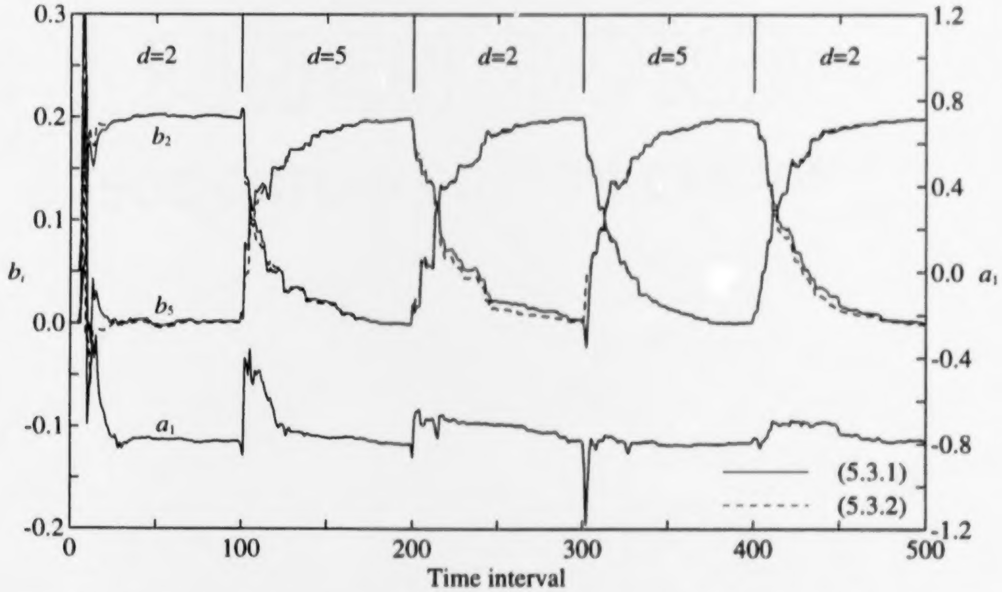


Figure 5.1: Recursion of extended numerator models to a time delay varying plant with open loop input excitation.

change at the 100th sampling interval. This has the effect of compensating for changes in plant time delay by adapting the time constant of the model. As a result, the b_i coefficients of (5.3.2) converge slightly faster than the b_i coefficients of (5.3.1). It is also interesting to note that the gain of (5.3.1) changes sign momentarily at the 303th sampling interval which causes an abrupt and incorrect change in the b_5 parameter. It was observed in closed loop experimental tests that effects such as colored noise, step disturbances and input-output data filtering can also bias the value of a_1. It is therefore established that (5.3.2) is superior to (5.3.1) for the purpose of time delay estimation.

The next step in the formulation of the ENR method is the extraction of a distinct time delay from B_d in (5.3.2). The value of b_i relative to ΣB_d reflects the correlation between the current plant output and the plant input $i+d_{min}$ intervals in the past. As was shown in Figure 5.1, the b_i parameter corresponding to the current plant delay converged approximately to the value of the plant numerator while all remaining b_i's converged very close to zero. Likewise, if the actual plant delay is not a multiple of the sampling interval, the two b_i's that straddle the actual delay will converge to non-zero values. For example, the discrete model structure

$$G\left(q^{-1}\right)=\frac{y(t)}{u(t-1)}=\frac{q^{-\ell+1}\left(b_1+b_2q^{-1}\right)}{1-a_1q^{-1}} \tag{5.3.3}$$

approximates the continuous plant structure

$$G(s)=\frac{e^{-(\ell T_s-mT_s)s}}{\tau s+1} \tag{5.3.4}$$

where ℓ is an integer >1, $1\geq m\geq 0$ and b_i's are non-zero. In addition, the relative weights of b_i can be used to approximate the fractional delay as given by

$$\frac{b_2}{b_1+b_2}=1-m \tag{5.3.5}$$

It is formally shown in the following lemma that as the sampling time approaches zero, the approximation in (5.3.5) becomes an equality.

lemma

$$\lim_{T_s \to 0} \frac{b_2}{b_1 + b_2} = 1 - m \tag{5.3.6}$$

Proof

The discrete approximation to (5.3.4) is given by the following modified z transform (Franklin *et al.*, 1990)

$$\begin{aligned} G(z) &= \frac{z-1}{z^{\ell+1}} \mathcal{Z}\left\{ \frac{e^{mT_s s}}{s} - \frac{e^{mT_s s}}{s + 1/\tau} \right\} \\ &= z^{-\ell} \frac{\left(1 - e^{-mT_s/\tau}\right)z + \left(e^{-mT_s/\tau} - e^{-T_s/\tau}\right)}{z - e^{-T_s/\tau}} \end{aligned}$$

or in terms of the backward shift operator, q^{-1}

$$G\left(q^{-1}\right) = q^{-\ell} \frac{\left(1 - e^{-mT_s/\tau}\right) + \left(e^{-mT_s/\tau} - e^{-T_s/\tau}\right)q^{-1}}{1 - e^{-T_s/\tau} q^{-1}}$$

$\lim_{T_s \to 0} \frac{b_2}{b_1 + b_2}$ can now be written as

$$\lim_{T_s \to 0} \frac{e^{-mT_s/\tau} - e^{-T_s/\tau}}{1 - e^{-T_s/\tau}} \tag{5.3.7}$$

The limit in (5.3.7) can be solved by applying l'Hospital's Rule

$$\lim_{T_s \to 0} \frac{e^{-mT_s/\tau} - e^{-T_s/\tau}}{1 - e^{-T_s/\tau}} = 1 - m$$

which concludes the proof of (5.3.6).

Increasing the sampling time is shown to decrease the accuracy of the approximation in (5.3.5) as shown in Figure 5.2. Although there is no consensus on the sampling time with respect to the dominant time constant of the process, a sampling time of τ/10 is included in the intervals of most of the published guidelines (Seborg *et al.*, 1989). The solution of (5.3.5) that corresponds to a sampling time of τ/10 results in a slightly underestimated but insignificant error as shown in Figure 5.2.

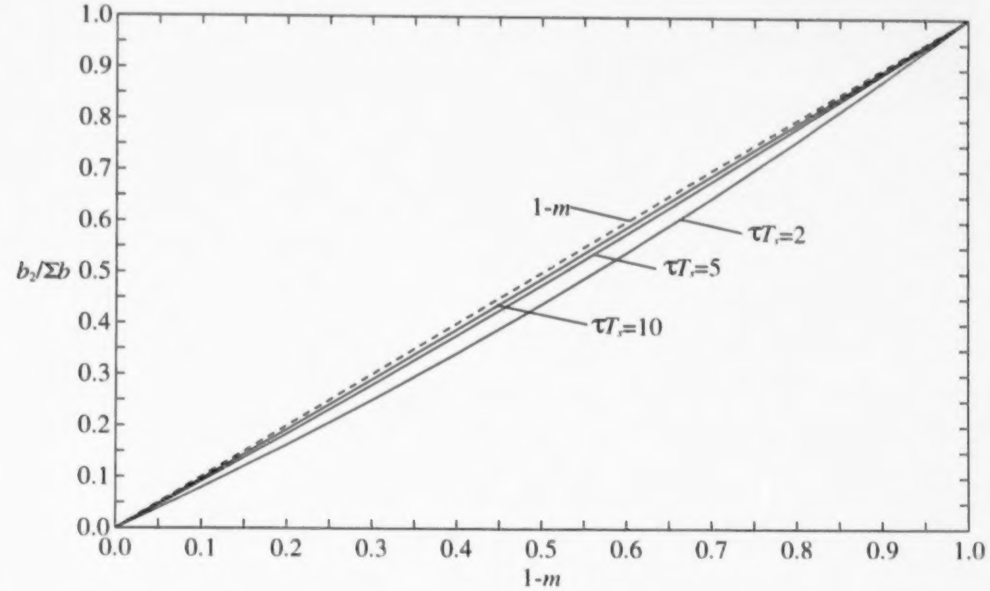


Figure 5.2: Graphical approximation of fractional time delay by comparing the relative weights of B_d.

The main idea behind the ENR method is to extend the concept of (5.3.5) to any order of B_d. This is accomplished by applying the method of moments whereby each delay index is weighted by the corresponding b_i coefficient. The delay estimate then corresponds to the zero moment position of the delay index or alternatively the weighted average of the delay index which can be expressed by

$$i^* = \frac{\sum_{i=1}^{nd} (i-1) b_i}{\sum_{i=1}^{nd} b_i} \tag{5.3.7}$$

where nd is $d_{\max}$-$d_{\min}$+1. The plant time delay estimate from the ENR method is then given by

$$\hat{d} = d_{\min} + i^* \tag{5.3.8}$$

which can take fractional values of the sampling interval. A simulated application of ENR is shown in Figure 5.3 for a first order plant with a variable time delay subject to white open loop excitation and low level noise. Recursive least squares (see Shah and Cluett, 1991 for details) with a constant forgetting factor of 0.95 was used to update the model:

$$y(t)-0.9048y(t-1)=\left(q^{-4}b_1+q^{-5}b_2+q^{-6}b_3+q^{-7}b_4+q^{-8}b_5+q^{-9}b_6\right)u(t-1)$$

at one second intervals and the fixed a_1 parameter was the same as the discrete plant. Although the time delay estimated by ENR is briefly erratic at the start of the run because the covariance matrix is not yet rich with plant data, the estimates converge quickly in response to changes in the plant time delay. At the 300th sampling instant, the plant time delay changes to 8.5 seconds which is not a multiple of the one second sampling time. Despite this, the ENR time delay estimate converges to the true plant time delay.

For implementation in the plant model (with A and B of the same order), the ENR result (5.3.8) must be rounded to the nearest integer. A first order plant model will be the most compatible with ENR estimates because ENR is based on a model with first order dynamics. As mentioned for the VRE method, this should not be a problem because the applications for which time delay estimation are most important are when the time delay is a dominant factor of the plant dynamics and a first order model is a sufficient approximation.

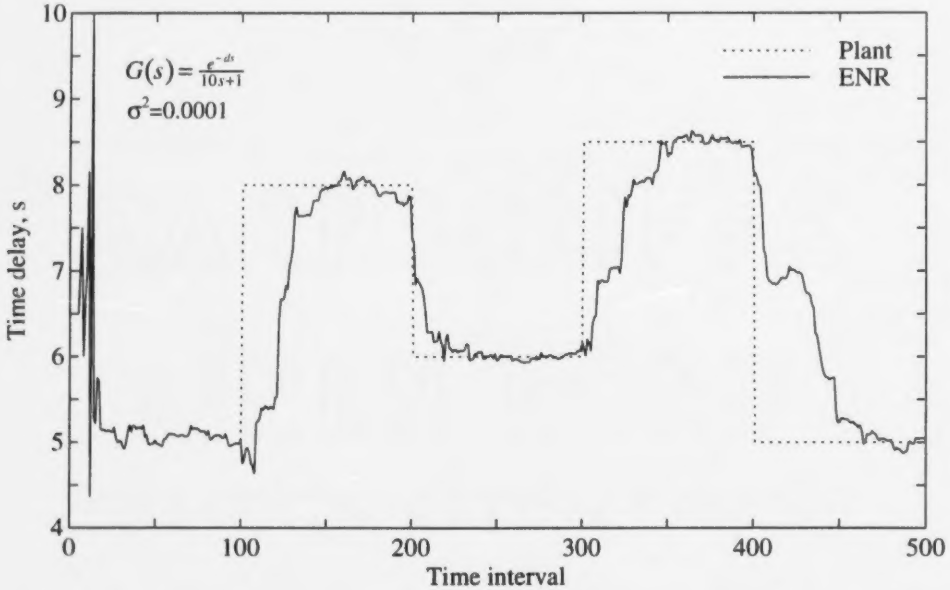


Figure 5.3: Application of ENR to a time delay varying first order plant subject to open loop excitation.

5.3.2 Statics Interpretation of ENR

A statics analogy of the ENR method is presented in this section in order to provide a different perspective of ENR. Consider a horizontal zero mass rigid member with a length of *nd*-1 units. Now imagine that each b_i coefficient is a vertical force positioned at *i*-1 units from the left edge of the member as depicted in Figure 5.4 for the b_i's: . 0.14, 0.17, 0.63, 0.11, 0.15. The ENR estimate of the time delay is the position of a fulcrum or bolt measured from the left that results in a zero moment of the rigid member. For this example, the zero moment position (and also the time delay estimate based on B_d) of the member is 1.97 units. This analogy holds irrespective of the sign (i.e. + or -) of the b_i coefficients.

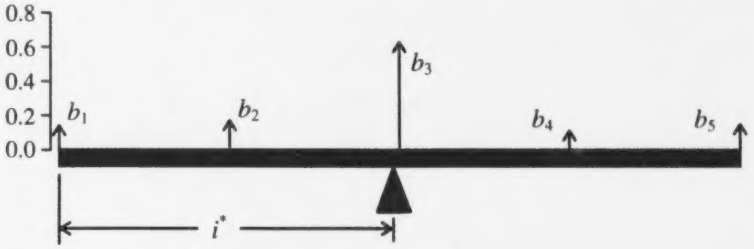


Figure 5.4: Statics analogy of ENR.

5.3.3 Practical Implementation of ENR

This section outlines the steps required for the recursive implementation of ENR based on closed loop input-output plant data. Recursive least squares based on an upper diagonal factorization (AUDI) (see Chapter 4 or Niu *et al.*, 1992 for a description of AUDI) method was chosen to update the ENR model (5.3.2) for several reasons. First, AUDI is numerically superior to classical RLS (i.e. without *UD* factorization), second, with little additional computational effort, AUDI provides estimates of the noise variance and parameter variance (Niu *et al.*, 1995) and third, stopping rules developed for AUDI prevent implementation of ENR when the current estimate is adequate. It was suggested by Shook *et al.* (1992) that band pass filtering of input-output data is recommended for use with least squares estimators such as AUDI to be compatible with long range predictive control laws. The block diagram representation of the on-line plant model adaptation is given in Figure 5.5 where $\frac{A}{C_e}$ is a bandpass filter. Outliers and bad data can be detected by comparing the current residual with the previous residual. If the difference

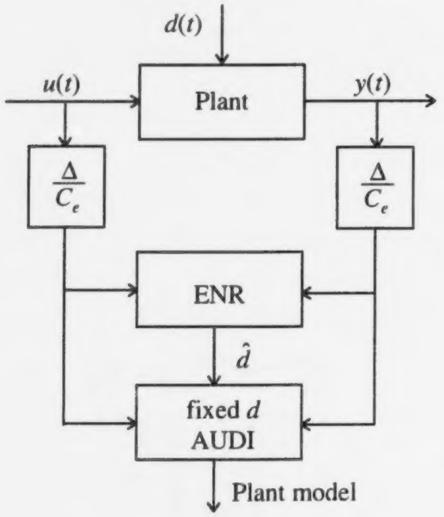


Figure 5.5: Block diagram representation of ENR in combination with plant model adaptation for an on-line implementation.

between the current and previous residual errors is greater than three times the standard deviation of the noise, then it can be concluded that the current input-output data are abnormal and should not be used for identification. Unnecessary updating of the time delay during steady state conditions is avoided by using the stopping rule developed by Niu (1994). In concept, the stopping rule discontinues the current update if the square of the current prediction error is less than the noise variance. Although not employed in the present work, the measure of identifiability conditions outlined by Niu *et al.* (1994) can also be used to determine if the current input-output data contains sufficient excitation for identification of the plant model. Filtering of the ENR estimate is strongly recommended to ensure smooth tracking of the plant time delay. The steps required to implement ENR based on recursive AUDI are outlined in an algorithmic form in Table 5.2. The noise variance can be estimated either by the variance of the residuals from batch data or on-line using the developments of Niu *et al.* (1994).

5.3.4 ENR Time Delay Uncertainty Estimate

Uncertainty bounds are an important indication of confidence and reliability in any measured or estimated quantity. Using the U D matrices in AUDI (see Table 5.2), the covariance matrix can be computed on-line without computing a matrix inverse as given by

Table 5.2: Implementation of ENR Based on Recursive AUDI.

$U(0)=I_{nd+1}$ $D(0)=10\cdot I_{nd+1}$ $\hat{\theta}(0)=[0 \quad \cdots \quad 0 \quad 1]^T_{nd+1}$	initialization only at startup if desired, U and $\hat{\theta}$ can be initialized to the nominal plant values
$\phi(t)=\left[u(t-d_{\max}-1) \quad \cdots \quad u(t-d_{\min}-1) \quad y(t)-\bar{a}y(t-1)\right]^T$	regressor
$f=U^T(t-1)\phi(t)$	innovation sequence
$g=D(t-1)f$	
$\lambda(t)=1-\dfrac{\left[\hat{\theta}^T(t-1)\phi(t)\right]^4}{N\sigma_\xi^4\left(1+f^Tg\right)}$	variable forgetting factor (Niu, 1994) N is the asymptotic data length σ_ξ^2 is the noise variance
if $\lambda(t)>1-\dfrac{1}{N\left(1+f^Tg\right)}$ do not proceed; use previous estimate of d; do not update U or D end	stopping rule (Niu, 1994)
$\beta_1=\lambda(t)$ for j=1 to m $\beta_{j+1}=\beta_j+f_jg_j$	
$D_{jj}(t)=D_{jj}(t-1)\beta_j/\beta_{j+1}/\lambda(t)$	update the D matrix
$\mu_j=-f_j/\beta_j$ $\nu_{jj}=g_j$ end for j=2 to m $\nu_{1:j-1,j}=\nu_{1:j-1,j-1}+U_{1:j-1,j}(t-1)\nu_{jj}$	
$U_{1:j-1,j}(t)=U_{1:j-1,j}(t-1)+\nu_{1:j-1,j-1}\mu_j$ end	update the U matrix
$\hat{\theta}(t)=U_{1:nd+1,nd+1}(t)$ $B_d=\hat{\theta}_{nd:1}(t)$	extract the model from U
$d^*=d_{\min}+\dfrac{\sum_{i=1}^{nd}(i-1)B_d(i)}{\sum_{i=1}^{nd}B_d(i)}$	this is the ENR time delay estimate
$d^f(t)=\alpha d^*+(1-\alpha)d^f(t-1)$ $\hat{d}(t)=\text{round}\left[d^f(t)\right]$	filter the ENR estimate (0<α<1) and round to the nearest integer

$$C(t)=UDU^T=\left[\sum_{j=1}^{t}\lambda^{t-j}\phi(j)\phi^T(j)\right]^{-1} \tag{5.3.8}$$

where C is the covariance matrix and λ is the forgetting factor. The covariance of each parameter estimate corresponds to the elements on the diagonal of (5.3.8). The variance of each b_i coefficient is then computed by the following expression

$$\sigma_{b_i}^2=\sigma_\xi^2 C(nd-i,nd-i) \tag{5.3.9}$$

With the assumption of normally distributed coefficients of B_d, a confidence interval for each b_i estimate is easily approximated by

$$b_{i,CI}=b_i \pm t_{C,N-1}\frac{\sigma_{b_i}}{\sqrt{N}} \tag{5.3.10}$$

where N is the length of the data window considered in the identification scheme and $t_{C,N-1}$ is the critical value from the t-distribution tables. If the noise variance, σ_ξ^2, is computed on-line within the AUDI framework (Niu *et al.*, 1995) then the t_C value corresponds to the data length in Table 5.2. If the noise variance is based on batch analysis of a large quantity of data then infinite degrees of freedom can be used to establish the confidence interval of b_i.

The confidence interval of the ENR time delay estimate corresponds to the minimum and maximum value computed from the confidence intervals of b_i. A nonlinear optimization of (5.3.7) is required to determine the maximum and minimum values subject to the b_i's constrained to their respective confidence intervals. Developing an analytical solution to this problem requires solving the Kuhn-Tucker conditions (Reklaitis *et al.* 1983) of (5.3.7) which turn out to be very messy even for a low order numerator. Non-linear optimization appears to be the only realistic approach for computing the uncertainty of ENR from the respective $b_{i,CI}$.

An alternative approach to estimating the uncertainty of the ENR estimate is through the propagation of variance of the b_i coefficients. In general, the variance of a non-linear function of several variables can be expressed as the propagation of variance and covariance of each variable given by

$$
\begin{aligned}
y &= f\left(x_1, x_2, \cdots, x_n\right) \\
\sigma_y^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial y}{\partial x_i}\right)\left(\frac{\partial y}{\partial x_j}\right)\sigma_{x_{i,j}}^2
\end{aligned}
\tag{5.3.11}
$$

where $\sigma_{x_{i,i}}^2$ is the variance of x_i and $\sigma_{x_{i,j}}^2$ is the covariance of x_i and x_j. The variance of the ENR delay can then be expressed as

$$
\sigma_d^2 = \sigma_\xi^2 \sum_{i=1}^{nd}\sum_{j=1}^{nd}\left(\frac{\partial i^*}{b_i}\right)\left(\frac{\partial i^*}{b_j}\right)C(nd-i, nd-j)
\tag{5.3.12}
$$

where

$$
\frac{\partial i^*}{\partial b_j} = \frac{(j-1)\sum_{i=1}^{nd} b_i - \sum_{i=1}^{nd}(i-1)b_i}{\left(\sum_{i=1}^{nd} b_i\right)^2}
\tag{5.3.13}
$$

Using (5.3.12) and (5.3.13), the confidence interval of the ENR time delay estimate can be determined in the same fashion as in (5.3.10). The estimation of the ENR uncertainty by (5.3.12) and (5.3.13) consists of simple multiplications which can be performed on-line with a low computational effort. In addition to the identifiability conditions mentioned in section 5.3.3, the ENR uncertainty estimate can be used to decide whether the current time delay estimate should be implemented in the plant model.

5.4 Simulation Study

The purpose of this section is to compare the performance of the VRE technique with the proposed ENR method for on-line estimation of time delay in closed loop simulation. Adaptive predictive PID (see Chapter 4 or Miller *et al.*, 1995) in combination with the discrete plant $G_p = \frac{q^{-d}0.1}{1-0.9q^{-1}}$ is used for all simulation runs.

5.4.1 Persistent Servo Excitation

Although persistent servo excitation is an unrealistic test for adaptive control, most of the literature in the area of on-line time delay estimation exclusively uses servo control without disturbances or noise. The simulation results in this section are not presented as an absolute evaluation of the ENR method but rather to compare it with the performance of the VRE technique for circumstances similar to those presented by Elnaggar *et al.* (1991). Predictive PID controller constants are based on the GPC parameters $N_1 = 1$, $N_2 = 30$ $N_u = 2$, $\lambda = 0.3$, $\gamma = 0.0$, $\gamma_y = 1.0$ and $C_C = 1\text{-}0.8q^{-1}$ while the input-output data is filtered by $\frac{\Delta}{1-0.05q^{-1}}$ for model estimation by both AUDI algorithms (plant model and ENR model). The data length (N) was 40 and the noise variance (σ_ξ^2) was 10^{-5} for computation of the variable forgetting factor used in plant model estimation. Figure 5.6 shows the ENR method in combination with adaptive predictive PID for a square wave setpoint input. The $\bar{a}_1$ coefficient in (5.3.2) was set to 0.9 while N and σ_ξ^2 in Table 5.2 were 40 and 10^{-10}, respectively. Estimation of the time delay by ENR responds to changes in the plant time delay very quickly with minimal mismatch as shown by the bottom

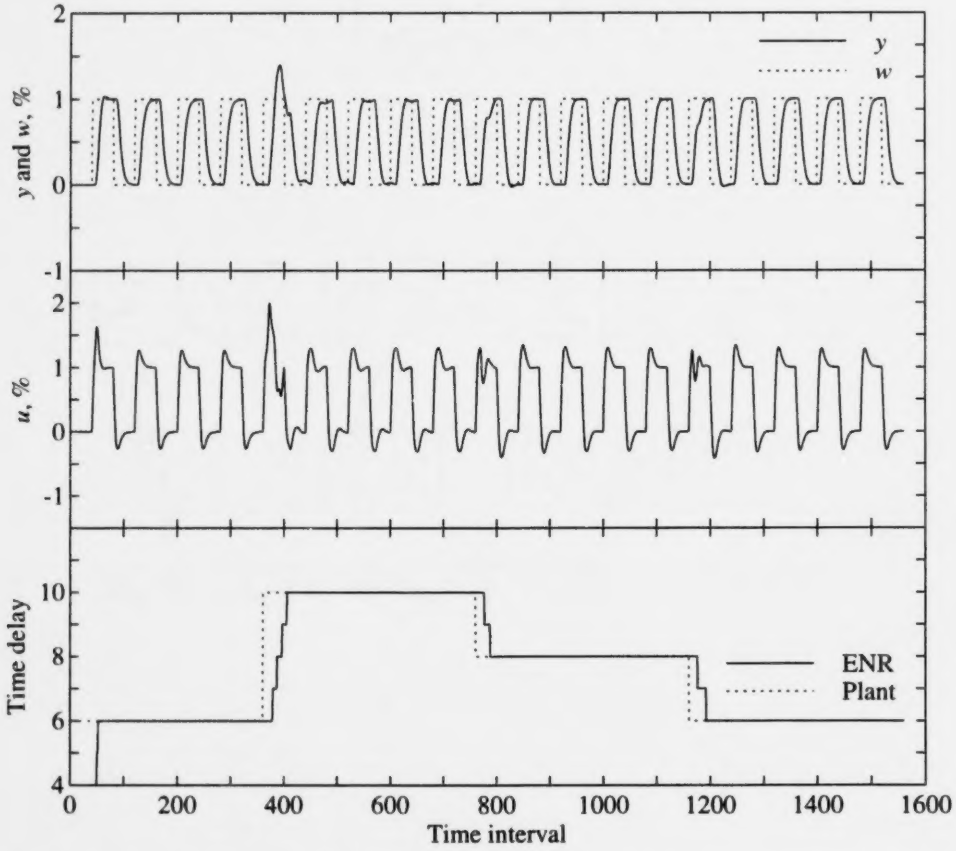


Figure 5.6: Adaptive predictive PID in combination with ENR subject to persistent servo excitation.

plot. Consequently, the response of the manipulated variable is excellent during changes in the plant time delay. The effect of a significant mismatch in $\overline{a}_1$ is shown in Figure 5.7 for the same setpoint changes and plant as in Figure 5.6. The trajectory of the ENR time delay estimation is skewed by +1 for $\overline{a}_1 = -0.85$ and -1 for $\overline{a}_1 = -0.95$. When a_1 is overestimated (underestimated), the time constant is overestimated (underestimated) so it is expected that the weighting of the numerator coefficients will reflect the time delay best suited to the ENR model and not the plant time delay. Despite this fact, it is recommended that a_1 be fixed for use with the ENR method because other factors such as colored noise and the selection of C_e have a significant influence on the convergence of a_1.

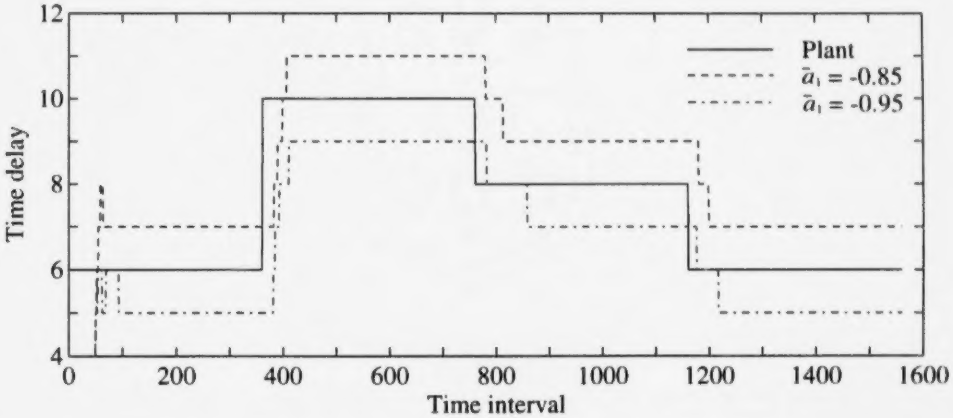


Figure 5.7: Effect of mismatch in *a* for closed loop on-line time delay estimation using ENR.

Time delay estimation using the VRE method with a forgetting factor of 0.99 is displayed in Figure 5.8 for the same plant and setpoint input as used for the ENR simulation. The VRE algorithm given in Table 5.1 was found to perform very poorly so the value of a_m in (5.2.8) was set to 0.9 instead of the recommended value of 1.0 by Elnaggar *et al.* (1991). With this modification, the trajectory of the VRE estimated time delay responds quickly to changes in the plant time delay although the estimates are very erratic. The simulation results reported by Elnaggar *et al.* (1991), based on a first order plant with a varying time delay, are not erratic as shown in Figure 5.8 so it is assumed that heavy filtering of the VRE time delay estimates was required although it is not included in the algorithm given in Table 5.1. The VRE time delay trajectories for a mismatch in *a* are displayed in Figure 5.9. Similar to the ENR results, the time delay estimates are skewed compared to Figure 5.8.

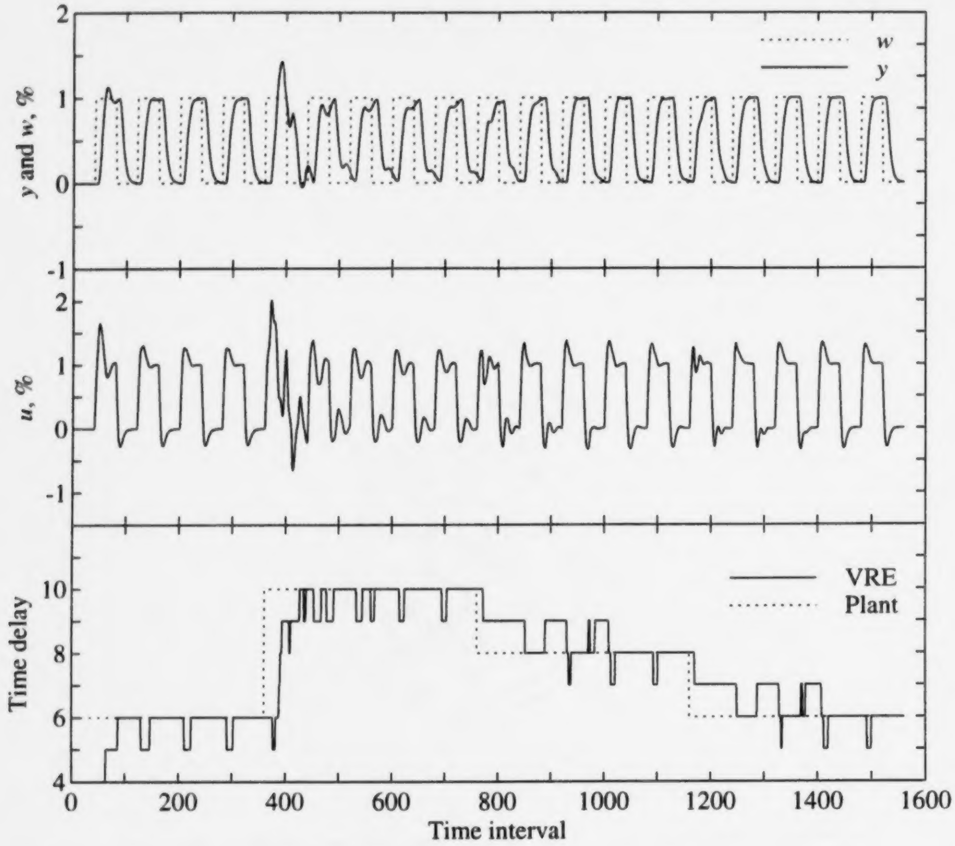


Figure 5.8: Adaptive predictive PID in combination with VRE subject to persistent servo excitation.

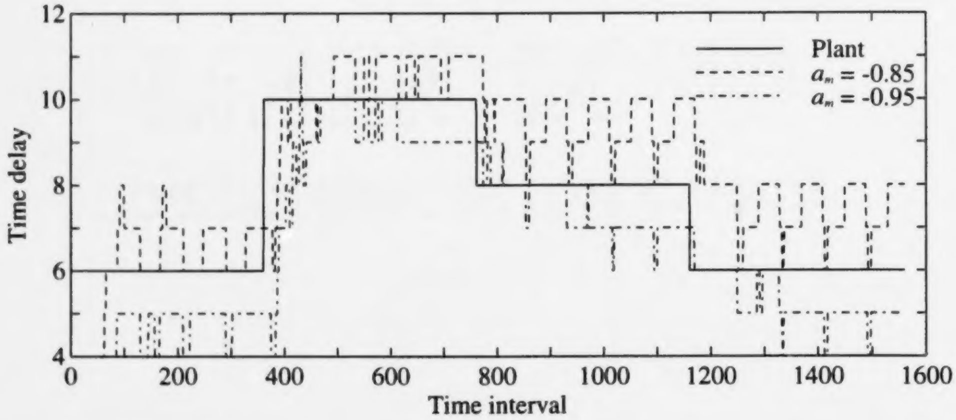


Figure 5.9: Effect of mismatch in *a* for closed loop on-line time delay estimation using VRE.

5.4.2 Regulatory Control

The main objective of chemical process control is to regulate a process in the presence of disturbances by manipulating one or more of the inputs to the process. Minimization of both the manipulated variable variance and the effect of disturbances to the process are key factors in selecting a control scheme while excellent performance during setpoint changes is seldom required or desired in industry. It is therefore important for consideration of adoption of new control techniques in the process industries that their performance is evaluated thoroughly and realistically. Simulation results as presented in this section provide an objective evaluation because the conditions are completely controlled.

It is the intent of this section to compare ENR with VRE for time delay estimation of a time delay varying process that is noisy and subject to frequent step disturbances. Each run starts with an initial setpoint pulse to initialize all of the recursive estimation algorithms to the nominal plant dynamics. The discrete plant is the same as in the previous section while the plant output is subject to colored noise with $\sigma_{\xi}^{2} = 0.001$ that is correlated by $C = \frac{1-0.8q}{A\Delta}$. The bandpass filter in Figure 5.5 is chosen to be consistent with C (i.e. C_e = $1\text{-}0.8q^{-1}$). The data length (N) in the forgetting factors used by the ENR algorithm and the plant model AUDI algorithm was 200 for all runs in this section. Following the initial setpoint pulse, a step disturbance of 0.1 is applied and removed every 80 sampling intervals.

The performance of the adaptive controller in combination with ENR for time delay estimation is shown in Figure 5.10. Although this simulation is considerably more challenging than in the previous section, the ENR time delay estimate responds quickly to changes in the plant delay and is usually within one sample interval of the actual time delay. In particular, the ENR estimated time delay responds very quickly to the plant time delay increase at the 280th sampling interval. This reaction is essential to avoid unstable closed loop behavior following a significant increase in the plant time delay. Excellent sensitivity to small changes in time delay is also demonstrated by the response of the ENR estimate after the 600th sampling interval. The response of the manipulated variable is very good as a result of the time delay adaptation.

Adaptive predictive control with VRE for time delay estimation of the same plant as in Figure 5.10 is shown in Figure 5.11. A forgetting factor of 0.995 was used for the VRE algorithm in Table 5.1 while the value of a_m in (5.2.8) was the same as the plant. As can be seen, the trajectory of the on-line VRE estimate is very erratic and does not appear to respond to

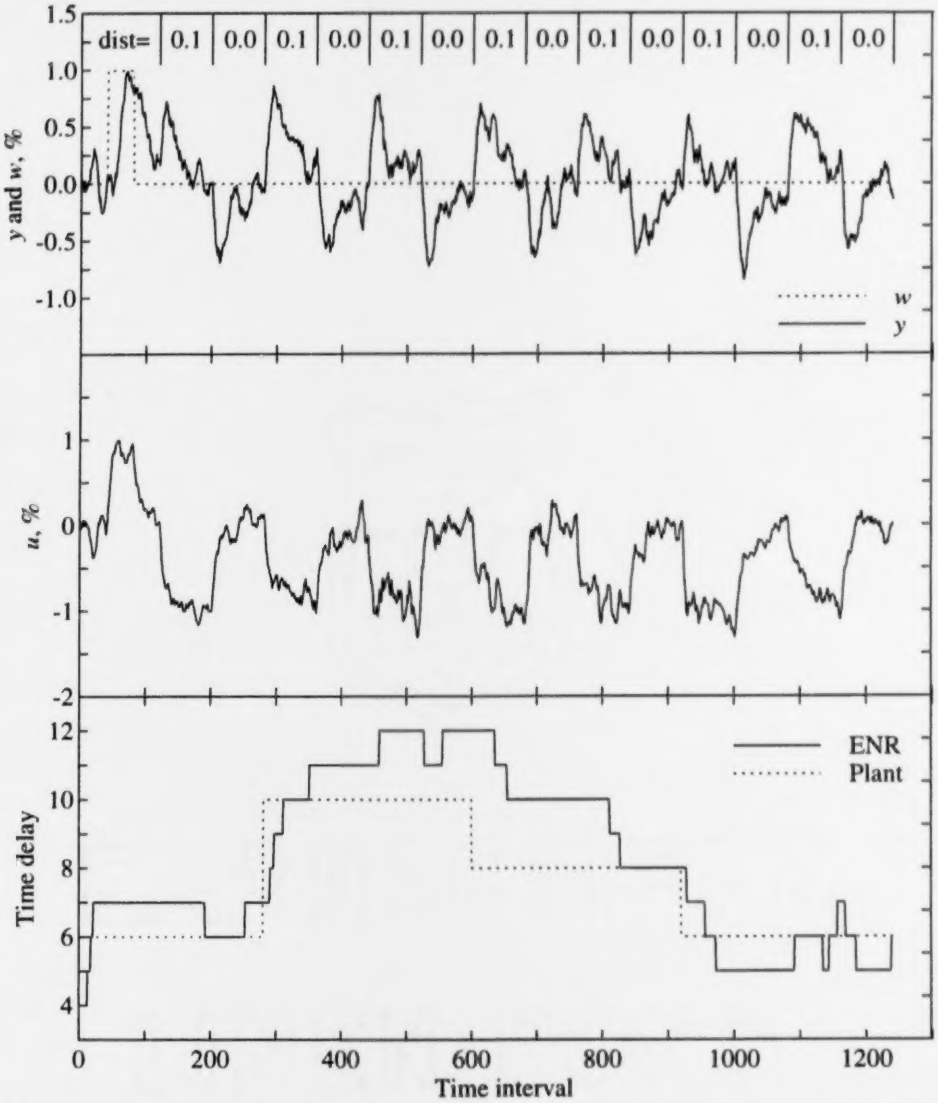


Figure 5.10: Adaptive predictive PID in combination with ENR for regulatory control of a noisy process subject to frequent disturbances.

changes in the plant time delay. The VRE time delay estimate climbed to over nine prior to the increase in plant time delay at the 380th sampling interval and remained above nine for the duration of the run. Response of the manipulated variable is stable but somewhat oscillatory because the time delay is significantly over estimated for most of the run. Because the VRE forgetting factor was 0.995, there is little hope in reducing the variation of the time delay estimates. It is clear from this example that the cross-correlation between input and output requires a high level of excitation to accurately estimate the time delay.

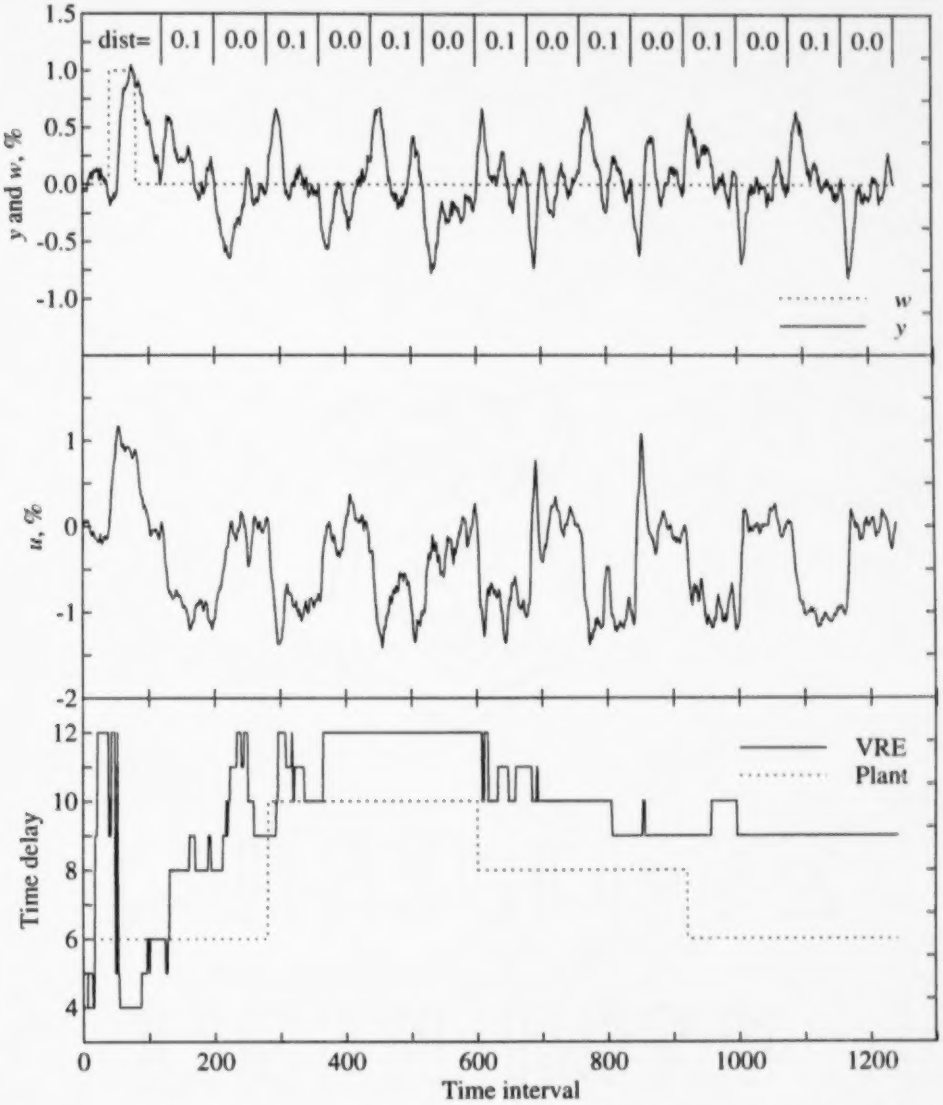


Figure 5.11: Adaptive predictive PID in combination with VRE for regulatory control of a noisy process subject to frequent disturbances.

5.5 Experimental Evaluation of ENR

Evaluation of adaptive control performance for control of a physical process is not only significantly more challenging but also more realistic than simulation of a process. Process noise, measurement noise, non-linearities, hysterisis and time varying process dynamics, typical of the environment encountered in controlling an industrial process are present. Although pilot scale processes are not subject to the same magnitude of these factors compared to an industrial process, a realistic evaluation can be performed.

The purpose of this section is to evaluate the performance of the ENR method in combination with adaptive predictive PID control of a steam heated stirred tank heater with a time varying time delay. Because the main objective of industrial control is regulation, the focus of this study is regulation during a series of disturbances and changes in plant dynamics.

5.5.1 Experimental Equipment

The schematic diagram of the stirred tank heater used in this study is shown in Figure 5.12. The tank consists of a double-walled glass vessel with an internal diameter of 14.5 cm and a height of 50 cm. Low pressure steam, manipulated by an equal percentage valve, is used to heat the fluid in the tank. Ordinary utility cold water is the process fluid which is mixed by a single impeller in the tank. Two temperature transmitters are located at 8.0 and 14.4 metres down stream from the tank exit which result in a time delay of 24 seconds and 44 seconds, respectively. An OPTO 22® I/O interface and a custom program written on the LabVIEW® software development system are used to control all of the manipulated variables and disturbances on the stirred tank heater.

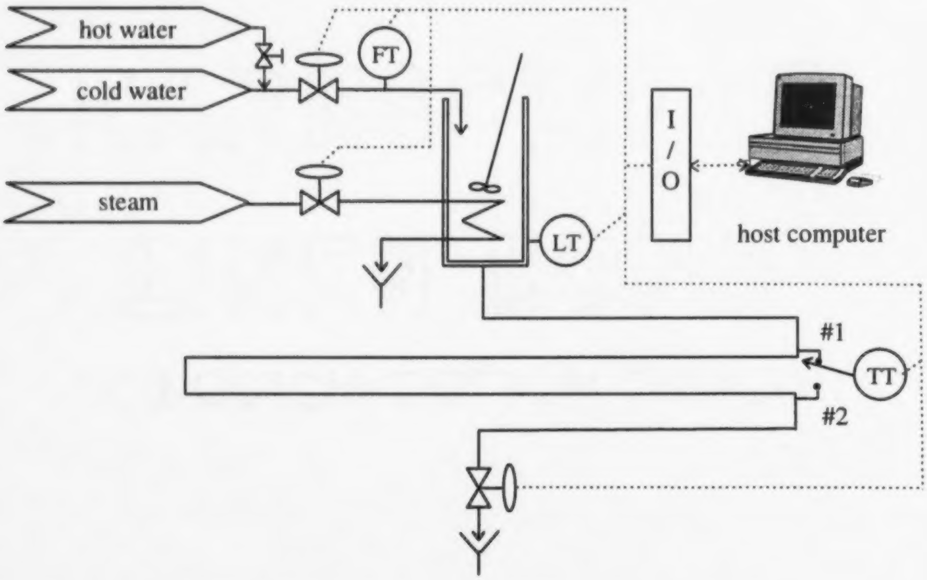


Figure 5.12: Schematic diagram of the stirred tank heater.

5.3.2 Implementation Details and Open Loop Modelling

The control configuration consisted of two single input single output loops for control of level and discharge temperature. A conventional PID controller was used to maintain the water level in the tank by manipulating the cold water control valve signal (cvs). Adaptive predictive PID with and without ENR was used to regulate the temperature of the water in the tank by manipulating the steam cvs. Steady state nominal operating conditions are given in Table 5.3.

Table 5.3: Nominal operating conditions for the stirred tank heater.

Cold water inlet temperature	10 °C
Tank discharge temperature	35 °C
Level	31 cm
Discharge cvs	100 %
Temperature transmitter position	#1

The two disturbances used in this study to alter the time delay in this study are a change in the discharge valve cvs and a change in the temperature transmitter position which both cause in a significant shift in the time delay. Open loop transfer functions for the nominal conditions in Table 5.3 in addition to the abnormal conditions which result from the disturbances used in this study are presented in Table 5.4. Changing the discharge cvs from 100 % to 75 % results in a severe upset to the discharge temperature and a significant shift in the dynamics of the stirred tank as indicated by the transfer function in Table 5.4. Switching the temperature transmitter position from #1 to #2 changes the time delay but does not disturb the discharge temperature.

Table 5.4: Open loop modelling of the stirred tank heater.

Deviation from Nominal Case	Continuous Transfer Function $\left(G_p = \frac{y(s)}{u(s)}, {}^{\circ}\!C/\%\right)$	Discrete Transfer Function $T_s = 4$ seconds
nominal	$\frac{0.381e^{-24s}}{48.3s+1}$	$\frac{q^{-6}0.0303}{1-0.921q^{-1}}$
discharge cvs=75%	$\frac{0.503e^{-36s}}{71.8s+1}$	$\frac{q^{-9}0.0273}{1-0.946q^{-1}}$
TT position # 2	$\frac{0.378e^{-44s}}{47.6s+1}$	$\frac{q^{-11}0.0305}{1-0.919q^{-1}}$

The sampling time for adaptive control was chosen to be four seconds based on the nominal stirred tank heater time constant of 48 seconds and the guidelines given in Seborg *et al.* (1989). An asymptotic data length of 150 was used in the forgetting factors for both AUDI algorithms (plant model and ENR model) while the ENR filter constant (α) in Table 5.2 was set to 0.2. Input-output data used by both AUDI algorithms was filtered by the band pass filter in Figure 5.5 with $C_e = (1\text{-}0.4q^{-1})^2$. For each run, an initial setpoint pulse is made to initialize the AUDI algorithms of the adaptive control schemes to the nominal plant dynamics. The last column of the U matrix was initialized to the nominal temperature dynamics listed in table 5.4. Furthermore, the value of a_1 was limited to a minimum of -0.995 by disregarding any a_1 estimate that is smaller than -0.995 in addition to disregarding updates of the U and D matrices. Controller constants of predictive PID for all runs are based on the GPC parameters $N_1 = 1$, $N_2 = 30$, $N_u = 2$, $\lambda = 0.8$, $\gamma = 0.3$, $\gamma_y = 1.0$ and $C_c = 1\text{-}0.8q^{-1}$ except the run in Figure 5.13 where $N_u = 1$.

5.5.3 Experimental Results

The experimental results presented in this section are intended to show the ability of adaptive predictive PID in combination with ENR relative to fixed time delay control structures for control of the stirred tank heater.

The first series of experiments shows the performance of each control scheme in response to a 600 second period of reduced discharge flow followed by a return to the nominal flow for 600 seconds. Predictive PID control of the discharge temperature during a disturbance to the discharge cvs is shown in Figure 5.13. Shortly after the discharge cvs changes from 100 %

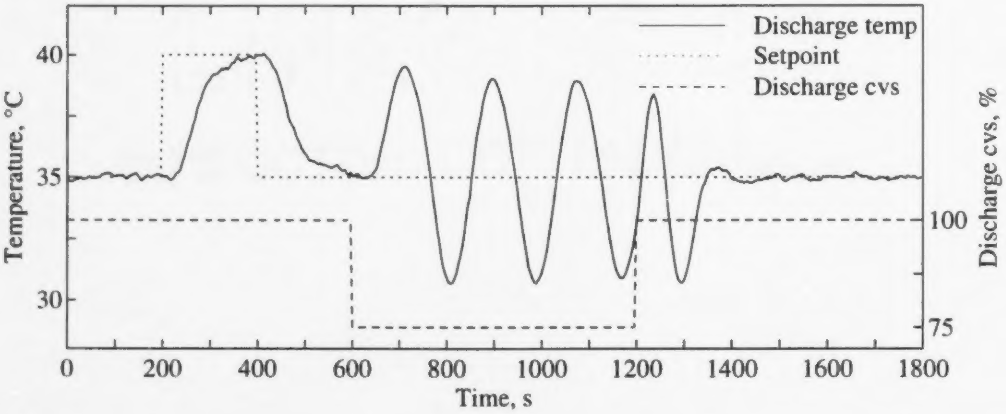


Figure 5.13: Fixed gain predictive PID control performance for a discharge flow rate disturbance.

to 75 %, the discharge temperature becomes marginally unstable which continues until 150 seconds after the discharge cvs returns to 100 %. As indicated by Table 5.4, the time delay increases by 50 % as a result of this disturbance. This performance is clearly unacceptable which proves that a long range predictive controller based on a fixed model is not suitable for control of processes with a significantly varying time delay. It is possible that detuning the predictive PID controller constants would result in stable temperature response for this type of disturbance. However, the overall performance would be very detuned. Adaptive predictive PID control for the same reduction in discharge flow rate employed for the test result in Figure 5.13 is displayed in Figure 5.14. Although the time delay is fixed, the time constant and gain of the model adapt to over estimated values in response to the time delay mismatch. This results in a stable but severely detuned response. In particular, the a_1 parameter estimate in Figure 5.14 converged very quickly to its minimum allowable value of -0.995 in response to the reduced flow disturbance introduced at 600 seconds. However, convergence of a_1 to its nominal value was very sluggish after the flow returned to its nominal value thus giving rise to the detuned temperature response. In addition, it was observed on subsequent runs without the imposed minimum value of a_1 that the gain of the model changed sign and the temperature response

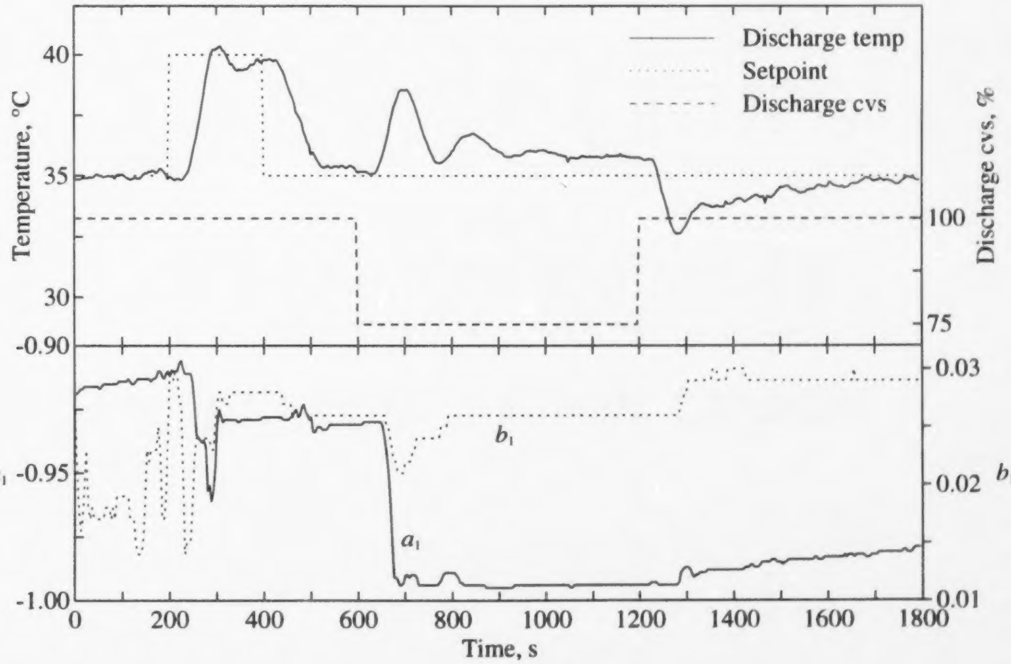


Figure 5.14: Adaptive predictive PID control performance for a discharge flow rate disturbance.

became unstable. It is uncertain whether this is a case specific behavior or if it applies to control of any process with a time varying delay. The temperature control that was achieved using adaptive predictive PID in combination with ENR is shown by the response displayed in Figure 5.15 for the same disturbance. In response to the decreased discharge flow rate at 600 seconds, the time delay estimated by the ENR method shifted to 12 intervals and remained constant for the duration of the disturbance. Consequently, the discharge temperature is stabilized quickly after a short period of diminishing oscillations. The time delay estimate did not return to its nominal value within 600 seconds after the discharge flow rate returned to its nominal value although the discharge temperature was stabilized quickly which considerably reduced the input excitation. Moreover, the model parameter estimates, a_1 and b_1, converged to -0.95 and 0.025 during the

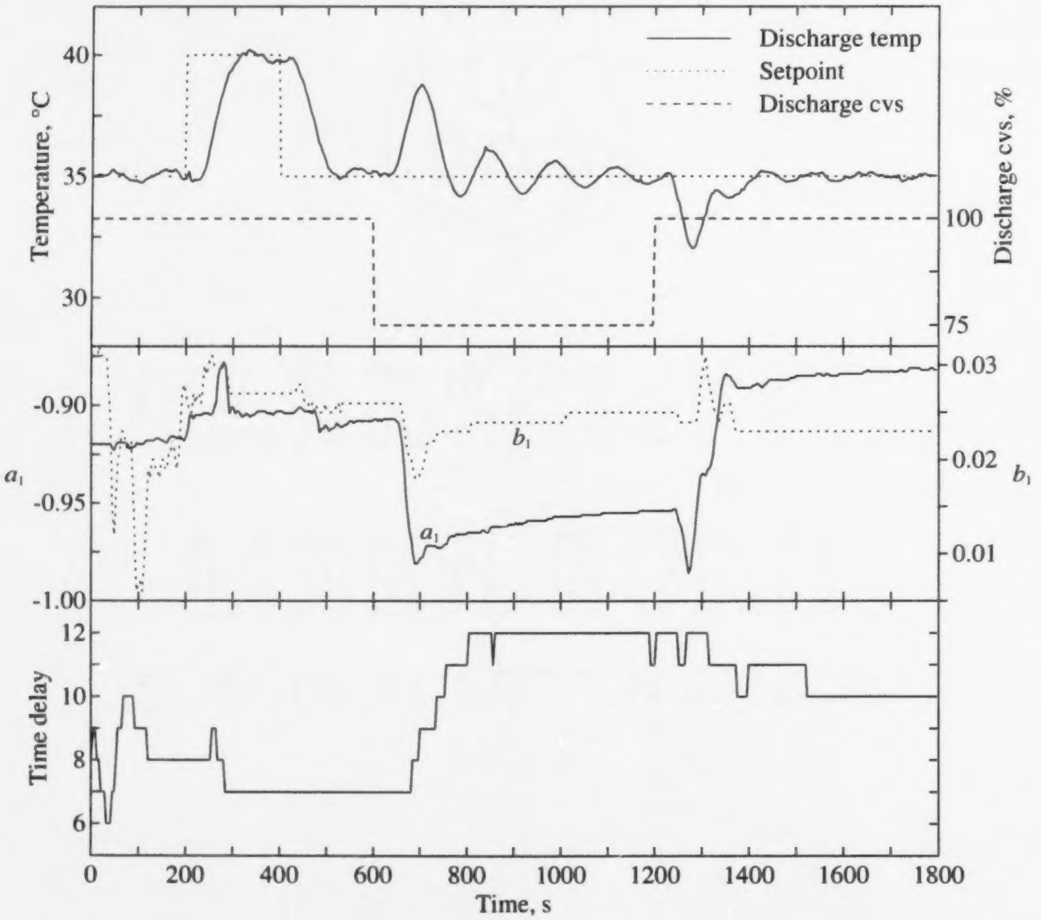


Figure 5.15: Adaptive predictive PID with ENR control performance for a discharge flow rate disturbance.

period of reduced discharge flow which compares very closely to the open loop estimates of -0.946 and 0.0273, respectively. The addition of ENR time delay estimation to adaptive predictive PID control is clearly effective for control of the stirred tank heater with a discharge flow rate disturbance.

The second series of tests is intended to test the ability of adaptive predictive PID with and without ENR to reject a disturbance following an increase in the plant time delay. Control performance of adaptive predictive PID is displayed in Figure 5.16 for a decrease in discharge flow rate at 600 seconds followed by an increase in the inlet water temperature from 10 °C to 17 °C at 1200 seconds. Similar to the response in Figure 5.14, the adaptive controller using a fixed time delay model is able to stabilize the discharge temperature following a decrease in discharge flow rate by overestimating the gain and the time constant of the process. However, as a result of the model plant mismatch, the controller is very detuned and cannot restore the discharge temperature to its setpoint even after 600 seconds from the time of the reduced discharge flow. In addition, the response to the inlet temperature disturbance is also very sluggish which results in a continued period of offset. A fixed time delay model is clearly inadequate for adaptive control of plants with a significantly time varying time delay.

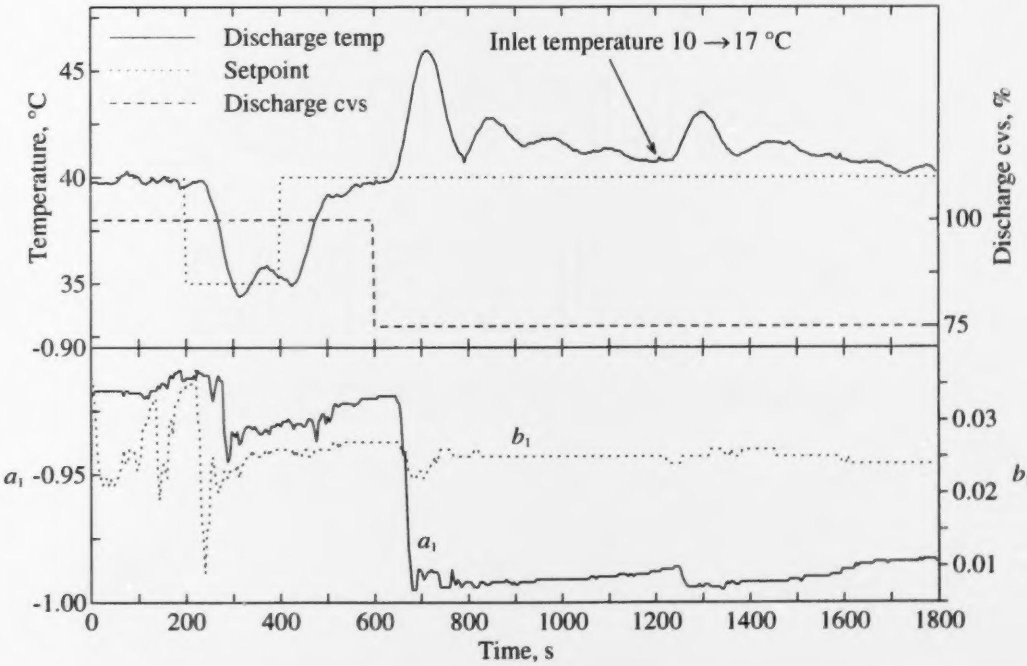


Figure 5.16: Adaptive predictive PID control performance for a discharge flow rate disturbance followed by an inlet temperature disturbance.

The temperature control behavior using adaptive predictive PID with ENR for the same disturbance pattern as used for the previous test is presented in Figure 5.17. As can be seen, the ENR time delay estimate increased from 7 to 11 sample intervals within 150 seconds of the decrease in discharge flow rate which resulted in the stabilization of the discharge temperature within 300 seconds of the disturbance. The plant model parameters converged almost exactly to the open loop values for the decreased discharge flow condition presented in Table 5.4. Consequently, the inlet temperature disturbance at 1200 seconds is rejected very quickly without oscillation. At the end of the test, the ENR time delay estimate was 10 sample intervals (40 seconds) which is only one sample interval larger than in the open loop model in Table 5.4. The response in Figure 5.17 shows that adaptive predictive PID with ENR is very capable of

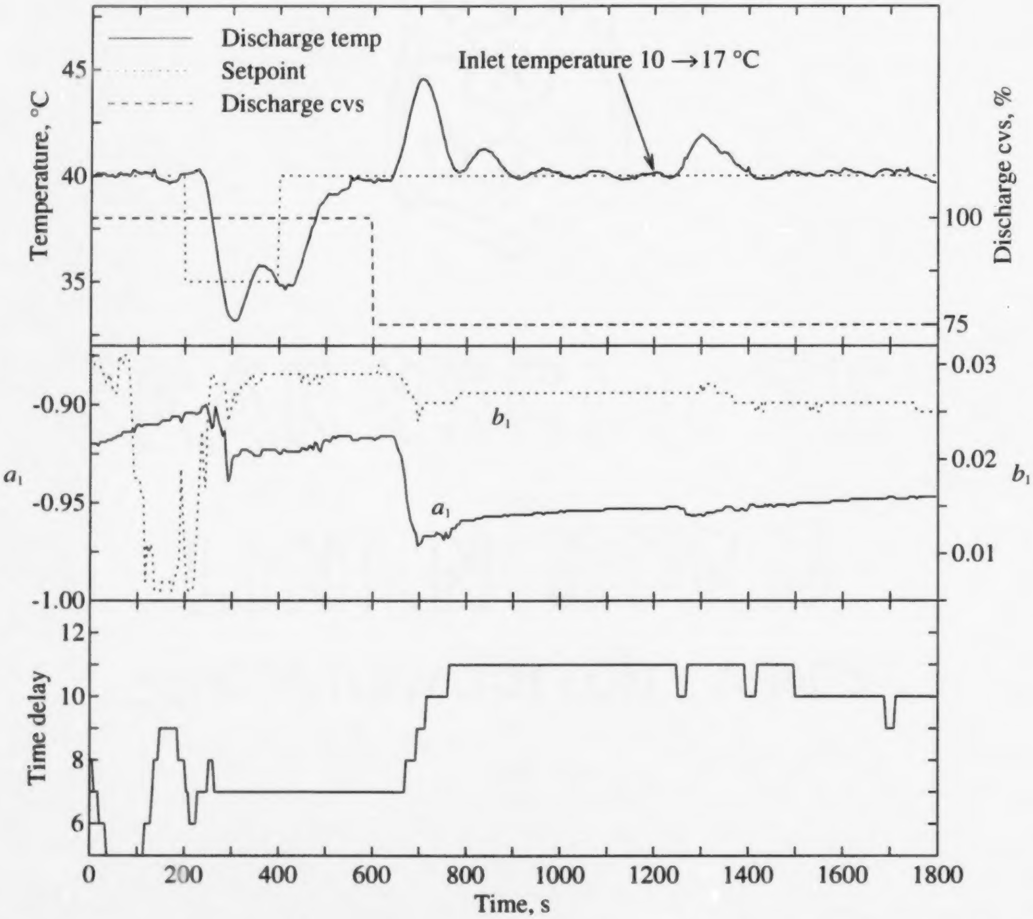


Figure 5.17: Adaptive predictive PID control performance for a discharge flow rate disturbance followed by an inlet temperature disturbance.

controlling processes subject to varying time delays and disturbances.

The third series of experiments is intended to show the performance of predictive PID and adaptive predictive PID with ENR for a process that exhibits a shift in time delay (i.e. plant gain and time constant remain constant). Performance of predictive PID, using a fixed model based on the nominal conditions, in response to switching the temperature transmitter position from #1 to #2 (see Figure 5.12) at 600 seconds is displayed in Figure 5.18. The inlet temperature was also perturbed at 600 seconds in order to accelerate the inevitable outcome of the response of the discharge temperature. This configuration resulted in an unstable discharge temperature response which demonstrates once again that use of a fixed time delay model with the predictive PID algorithm is inadequate for variable time delay processes. The control performance obtained using the adaptive predictive PID with ENR algorithm for the disturbance pattern is displayed in Figure 5.19. The time delay estimated by the ENR method responded much slower to an increase in the plant time delay compared to the discharge flow disturbance in Figures 5.17 and 5.15. This can be explained by the reduced level of input-output excitation compared to a reduced discharge flow rate. However, the estimated time delay eventually climbs to 10 sampling intervals which stabilizes the discharge temperature by 1400 seconds. The a_1 parameter converges to about -0.975 which is significantly different than the open loop estimate of -0.919. The likely reason for this difference is the extended period of time delay mismatch between the plant and the model while the ENR estimate is converging. In contrast, the b_1 parameter in the plant model remained relatively constant after the setpoint pulse which

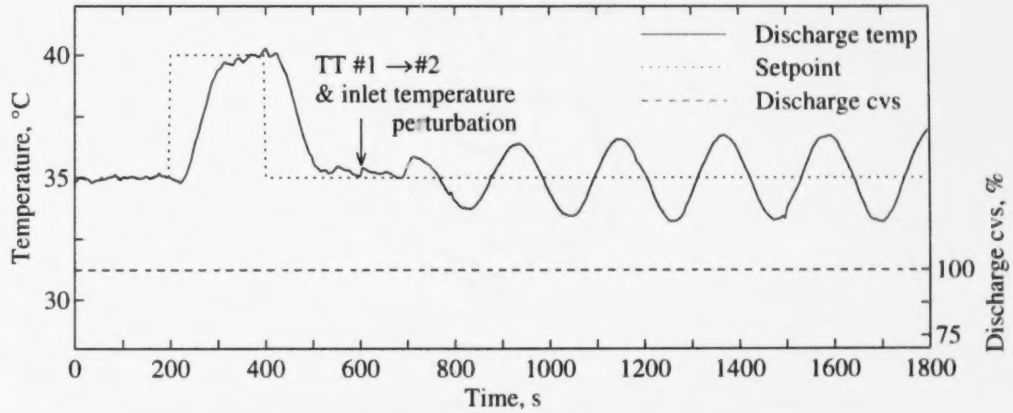


Figure 5.18: Predictive PID control performance for a shift in the plant time delay and a perturbation in the inlet temperature.

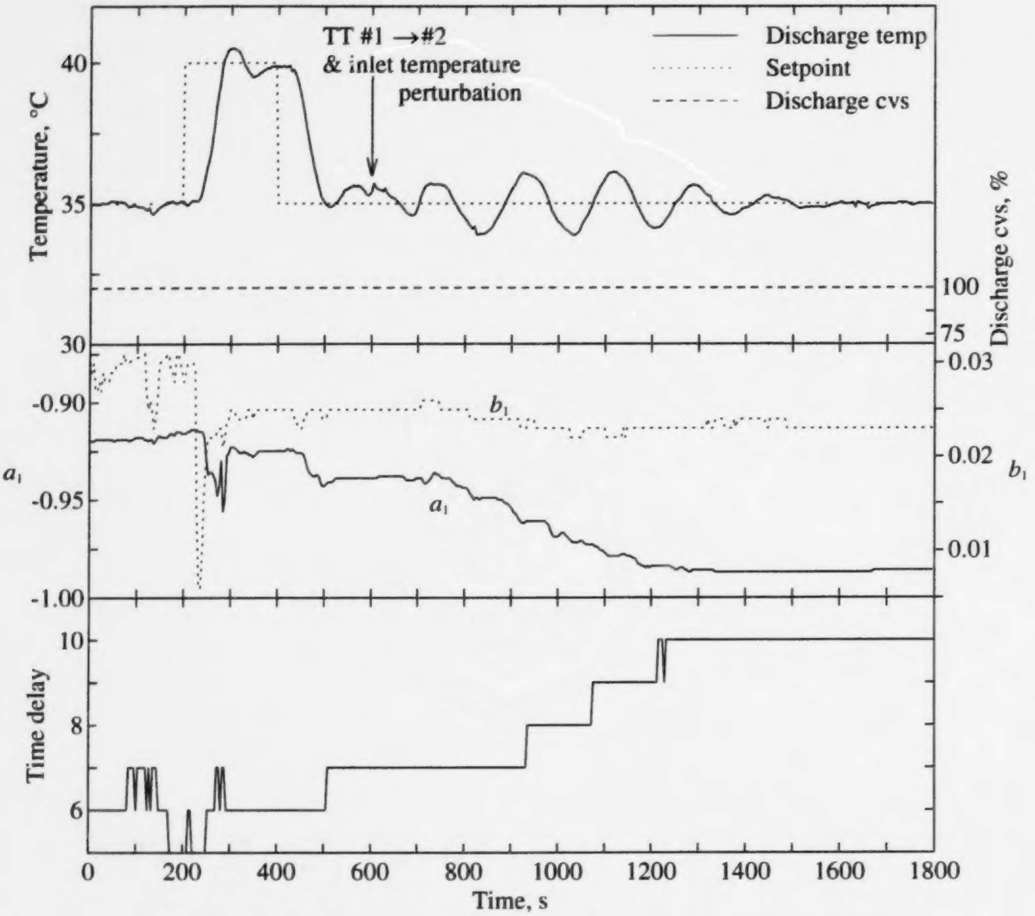


Figure 5.19: Adaptive predictive PID with ENR control performance for a shift in the plant time delay and a perturbation in the inlet temperature.

corroborates the decision to fix a_1 in the ENR model. However, it is important to note that in the absence of rich excitation, the ENR method produced constant time delay estimates rather than erratic estimates. This result motivated a more challenging experiment in which the temperature transmitter position is switched without an inlet temperature perturbation. The resulting control performance of adaptive predictive PID with ENR is displayed in Figure 5.20. In the presence of very low level input excitation, the ENR estimate does converge to 10 sampling intervals which also stabilizes the discharge temperature. It is not surprising that the value of a_1 deviates farther from the open loop estimate and the response of ENR is slower compared to the run in Figure 5.19 because the conditions for identification are even more challenging.

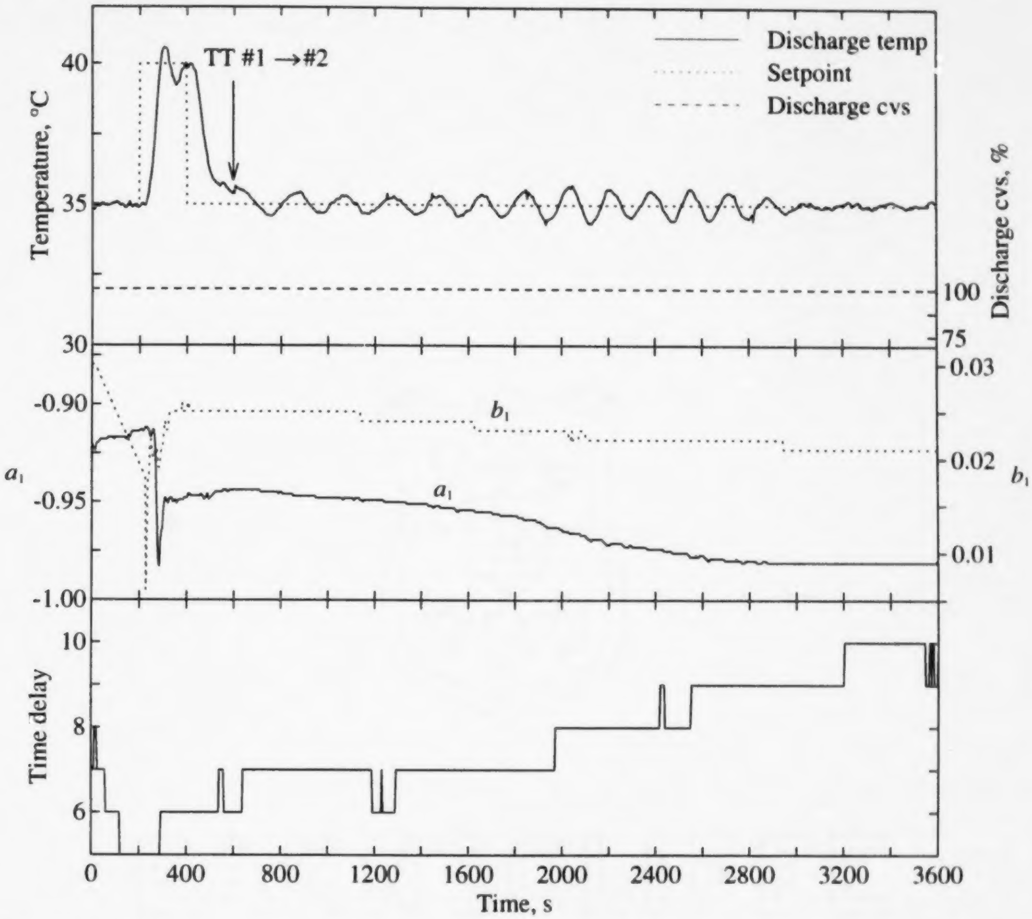


Figure 5.20: Adaptive predictive PID with ENR control performance for a shift in the plant time delay without an inlet temperature perturbation.

5.6 Conclusions

- The proposed ENR method for on-line estimation of time delay was shown to produce accurate estimates during regulatory adaptive control in simulation and experimental tests. ENR time delay estimates converged quickly in response to changes in the plant time delay which resulted in excellent closed loop performance in cases where controllers based on a fixed time delay model produced unstable or poor control performance.
- During periods of low plant input excitation, the ENR time delay estimates converged slowly but smoothly to the plant time delay in experimental tests.
- It is recommended that the value of the a_1 parameter of a first order extended numerator model be fixed to the nominal value for use with the ENR time delay estimation method.
- The ENR method was shown to be superior to the VRE method of Elnaggar *et al.* (1991) during regulatory control in simulation.

References

Åström, K.J., and T. Hägglund, *Automatic Tuning of PID Controllers*, ISA, NC, 1988.

Banerjee, P, and S.L. Shah, "Simultaneous estimation of gains and model orders of orthonormal function models using the augmented upper diagonal identification (AUDI) method," *submitted to IEEE trans. Signal processing*, 1995.

Chien, I.L., D.E. Seborg and D.A. Mellichamp, "A self-tuning controller for systems with unknown or varying time delays," *Proc. American Control Conf.*, pp 905-912, 1984.

Clarke, D.W. and C. Mohtadi, and P.S. Tuffs, "Generalized predictive control - part I. the basic algorithm," *Automatica*, Vol. 23, No. 2, pp 137-148, 1987.

Clough, D.E., and S.J. Park, "A novel dead-time adaptive controller," ," *IFAC Workshop on Adaptive Control of Chemical Processes*, pp 21-26, 1985.

De Souza, C.E., G.C. Goodwin, D.Q. Mayne and M. Palaniswami, "An adaptive control algorithm for linear systems having unknown time delay," *Automatica*, **24**, pp 327-341, 1988.

Dumont, G.A., A. Elnaggar and A. Elshafei, "Adaptive predictive control of systems with time-varying time delay," *Int. J. Adaptive Control and Signal Processing*, Vol. 7, pp 91-101, 1993.

Elnaggar, A., G.A. Dumont and A.L Elshafei, "Delay estimation using variable regression," *Proc. American Control Conf.*, Vol. 3, pp 2812-2817, 1991.

Ferretti, G., C. Maffezzoni and R. Scattolini, "Recursive estimation of time delay in sampled systems," *Automatica*, Vol. 27, No. 4, pp 653-661, 1991.

Franklin, G.F., J.D. Powell and M.L. Workman, *Digital Control of Dynamic Systems*, 2nd Ed., Addison-Wesley, Reading, MA, 1990.

Gabay, E. and S.J. Merhav, "Identification of linear systems with time-delay operating in a closed loop in the presence of noise," *IEEE Trans. Automatic Control*, **AC-21**, pp 711-716, 1976.

Gawthrop, P.J., and M.T. Nihtila, "Identification of time delays using a polynomial identification method," *Systems and Control Letters*, **5**, 267-271, 1985.

Kurtz, H., and W. Goedecke, "Digital parameter-adaptive control of processes with unknown dead time," *Automatica*, Vol. 17, No. 1, pp 245-252, 1981.

Miller, R.M., S.L. Shah and R.K. Wood, "Development of a stochastic predictive PID controller," *to be presented at the American Control Conf.*, Seattle, 1995.

Niu, S., D.G. Fisher and D. Xiao, "An augmented UD identification algorithm," *Int. J. Control*, Vol. 56, No. 1, pp 193-211, 1992.

Niu, S. and D.G. Fisher, "Monitoring parameter identifiability during on-line identification," *presented at the 3rd IEEE Conf. Control Applications (also submitted to Int. J. Cont.)*, Glasgow, August, 1994.

Niu, S. and D.G. Fisher, "Simultaneous estimation of process parameters, noise variance and signal-to-noise ratio," *IEEE Trans. Signal Processing*, in press, 1995.

Niu, S., "Augmented UD identification for process control," *Ph.D. thesis*, University of Alberta, 1994.

Pearson, A.E. and C.Y. Wuu, "Decoupled delay estimation in the identification of different delay systems," *Automatica*, **20**, pp 761-772, 1984.

Prasad, C.C., V. Hahn, H. Unbehauen and U. Keuchel, "Adaptive control of a variable dead time process with an integrator," *IFAC Workshop on Adaptive Control of Chemical Processes*, pp 71-75, 1985.

Rao, G.P., and L. Siuakumar, "Identification of time lag systems via Walsh functions," *IEEE Trans. Automatic Control*, **AC-24**, 806-808, 1979.

Reklaitis, G.V., A. Ravindran and K.M. Ragsdell, *Engineering Optimization: Methods and Applications*, pp 191-212, John Wiley and Sons, New York, 1983.

Robinson, W.R. and A.C. Soudack, "A method for the identification of time delays in linear systems," *IEEE Trans. Automatic Control*, **AC-15**, pp 97-101, 1970.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp, *Process Dynamics and Control*, pp 272-309, John Wiley and Sons, New York, 1989.

Shah, S.L. and W.R. Cluett, "Recursive least squares based estimation schemes for self-tuning control," *Can. J. Chem. Eng.*, Vol. 69, pp 89-96, 1991.

Shook, D.S., C. Mohtadi and S.L. Shah, "A control relevant identification strategy for GPC," *IEEE Trans. Automatic Control*, Vol. 37, No. 7, pp 975-980, 1992.

Zervos, C.C. and G.A. Dumont, "Deterministic adaptive control based on Laguerre series representation," *Int. J. Control*, **48**, pp 2333-2359, 1988.

Chapter 6

LabVIEW® for Experimental Process Control

Assessment of the LabVIEW software development system for the experimental implementation and evaluation of advanced process control is presented in this chapter. Comments on the development of specific functions required for execution of adaptive long range predictive control (LRPC) are given in addition to recommendations for future development.

6.1 Introduction

Experimental evaluation will always be an important component of process control research. Since most research occurs at the university level, it is essential that a cost effective platform be available for implementation of advanced process control and identification algorithms on pilot scale processes. It is equally important that control algorithms be easily implemented and modified. LabVIEW (this strange capitalization is a trademark of the manufacturer; Anon, 1993) is one such software development system that meets the above criteria and was subsequently used for the experimental implementation of adaptive predictive PID presented in earlier chapters.

The main purpose of this chapter is to specifically address the most challenging issues in the LabVIEW implementation of adaptive LRPC for control of pilot scale processes. There were three LabVIEW version upgrades (2.5.2, 3.0 and 3.0.1) during the course of this research and at the time of writing version 3.1 was released which indicates that LabVIEW is evolving at a rapid pace. This chapter is therefore not intended to function as a users manual for the control programs developed in this thesis or to evaluate LabVIEW relative to alternative software because this would quickly become obsolete. However, general recommendations are made regarding the role of LabVIEW in the future development of advanced process control applications.

6.2 General LabVIEW Description and Comments

6.2.1 General Description

LabVIEW is a general purpose software development system intended for laboratory data acquisition and control that is presently available for the PC (Windows), McIntosh, Sun and Hewlett Packard computer hardware. A graphical object oriented language is used to create programs which visually resemble electronic circuit diagrams. There are three main components of a LabVIEW program: the **block diagram** which is the source code, the **front panel** which is the graphical users interface and the **icon and connector** which is used to pass parameters between subprograms. Since LabVIEW programs are intended to emulate physical instruments, they are called virtual instruments. Libraries of some basic and some advanced functions are available for creating programs some of which are further discussed in section 6.3 relative to the implementation of adaptive LRPC. Virtual instruments are naturally modular in design whereby complicated tasks are easily subdivided into subtasks. Data acquisition utilities and the graphical users interface are particular strengths of LabVIEW and will be further discussed in section 6.3. Readers are urged to consult the LabVIEW users manual (Anon, 1993) for a more detailed description.

6.2.2 LabVIEW Programming Paradigms

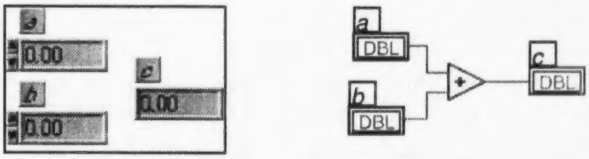
Perhaps the most significant difference between LabVIEW and conventional text based programming languages is the process of translating ideas or algorithms into executable code. When one is familiar with a conventional programming language (e.g. Pascal), learning another conventional language (e.g. C) requires only a relatively small amount of effort because the programming methodologies are directly transferable. Like all languages, LabVIEW has strict rules or syntax regarding the construction and execution of programs. Although LabVIEW contains many of the same structures found in conventional programming languages (e.g. while loop, for loop, case structure etc.), a very different methodology is required to solve problems. There is generally no preset order of execution except that a function can not execute until its input parameters are defined which is very different from a text based program. Consequently, the initial learning curve was found to be very steep by the author and also by undergraduate students in a real time computing course at the University of Alberta. However, after this initial learning period, construction of programs was perceived by the author to require approximately the same amount of time and effort as text based programming languages. One interesting observation was that people proficient in writing text based programs generally had a higher level of frustration during this initial learning period compared to inexperienced programmers.

6.2.3 Simple Examples

Two simple examples are described in this section to illustrate some programming paradigms of LabVIEW.

Example 1

Consider the most simple of problems; that is, adding two scalar numbers. In almost all programmatic languages, the entry $a+b=c$ will give the correct result. The LabVIEW front panel and block diagram for this problem is shown in Figure 6.1. It was required to create a front panel object for the inputs a and b and the sum, c which are automatically linked to their respective source or sink in the block diagram. A wiring tool is then used to connect these objects to the *add* function. This may seem at first to be a somewhat tedious procedure just to add two numbers. However, the above program can be run continuously and the inputs can be changed interactively without any additional effort. This latter feature would require considerable additional effort with any text based language.



Front Panel

Block Diagram

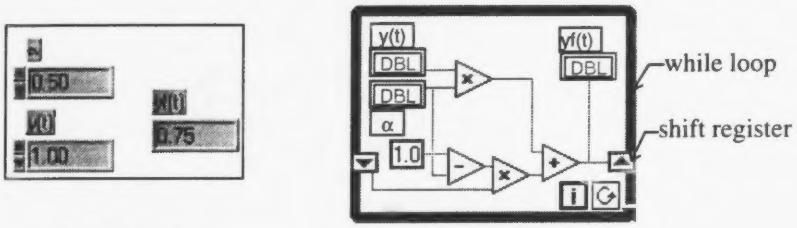
Figure 6.1: LabVIEW program to interactively add two numbers.

Example 2

The next example considers one possible LabVIEW implementation of a unity gain first order digital filter given by

$$y^{f}(t)=\frac{\alpha}{1+(\alpha-1)q^{-1}}y(t) \tag{6.2.1}$$

Figure 6.2 shows the LabVIEW front panel and block diagram for the implementation of (6.2.1).



Front Panel Block Diagram

Figure 6.2: LabVIEW implementation of a digital first order filter.

The large rectangular box in the block diagram of Figure 6.2 is a *while loop* and the arrow blocks located on the wall of the *while loop* represent a *shift register* which functions as a backward shift operator. In this fashion, $y^f(t\text{-}1)$ is available for computation of $y^f(t)$. Similar to Example 1, α and $y(t)$ can be changed interactively while the program is running or alternatively, $y(t)$ can function as an input parameter if the above program is used as a subprogram. From a text based language programmers point of view, this is not an obvious implementation of a digital filter although it is not particularly difficult or complicated.

6.3 Adaptive LRPC Implementation Issues

This section gives the author's personal perspective and assessment regarding some of the issues in the LabVIEW implementation of adaptive LRPC for control of the steam heated stirred tank heater presented in Chapters 4 and 5.

6.3.1 Program Structure

The highest level block diagram for the adaptive predictive PID program for control of the stirred tank heater is shown in Figure 6.3. Each shaded square in Figure 6.3 represents a subprogram, some of which contain up to ten lower levels of subprograms, while the thick shaded lines that

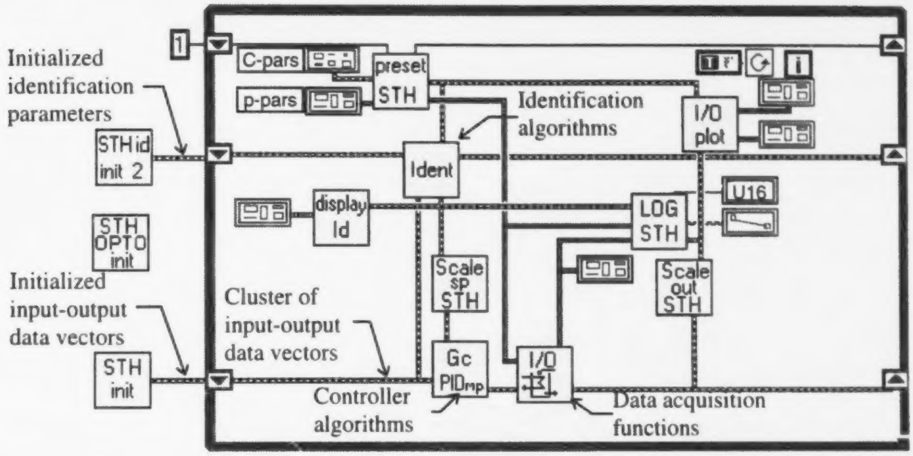

Initialized identification parameters
STH id init 2
STH OPTO init
Initialized input-output data vectors
STH init
1
C-pars
p-pars
preset STH
Identification algorithms
Ident
I/O plot
display Id
LOG STH
U16
Scale sp STH
Scale out STH
Cluster of input-output data vectors
Gc PIDmp
I/O
Controller algorithms
Data acquisition functions

Figure 6.3: Block diagram of adaptive predictive PID for control of a steam heated stirred tank heater.

connect the subprograms are data clusters containing scalars, vectors and matrices. Logical groups of functions are combined so that the general structure of the program is clearly visible. This object oriented approach in LabVIEW was found to be excellent for implementation of adaptive LRPC while troubleshooting was quickly isolated to particular modules.

6.3.2 Diophantine Identity

Solutions of two Diophantine identities over the prediction horizon are required for long range predictions in the LRPC law (see Chapter 2 for more details). Deconvolution and simple linear algebra operations, found in the advanced analysis libraries of LabVIEW, were used for the solution of both Diophantine identities. This proved to be the most challenging LRPC subfunction to implement in LabVIEW and required more time and effort than required for the Matlab® (Anon, 1994) implementation. One of the difficulties in the LabVIEW implementation was the result of the deconvolution function. When the last element of the current deconvolution solution is zero, the vector is truncated such that the last element is nonzero. Addition or subtraction of two unequal length vectors in LabVIEW is a valid operation and results in a truncated solution (e.g. [1 3 2 9 8]+[2 5 3]=[3 8 5]). Conversely, in Matlab this is an illegal operation. It was therefore necessary to pad the deconvolution solution with zeros to retain the correct vector length so that subsequent operations on the deconvolution result were correct (e.g. [1 3 2 9 8]+[2 5 3 0 0]=[3 8 5 9 8]).

6.3.3 GPC Control Law

Computation of the future control vector simply requires matrix multiplications, a matrix inverse and subtractions. There were no major problems with the LabVIEW implementation although each operation required a separate icon which resulted in a cluttered block diagram. In comparison, only one text line is required for the Matlab solution.

6.3.4 Augmented *UD* Identification (AUDI) Algorithm

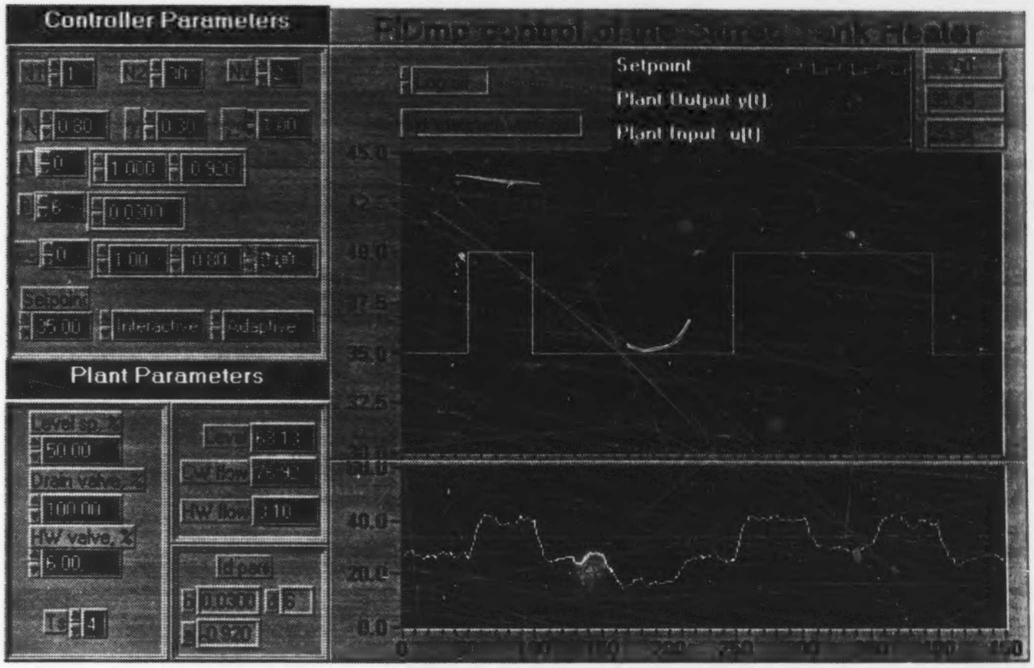
The algorithmic representation of AUDI given in Table 4.1 was implemented in LabVIEW for on-line estimation of the plant model which was used in conjunction with LRPC. Execution of the AUDI algorithm requires many matrix manipulations which can be very tedious and are complicated procedures in LabVIEW. In particular, replacing a row of a matrix with another row requires a "for loop" or a "while loop" to replace each element one by one. This is a single line operation in Matlab. Other difficulties in LabVIEW include redefining a matrix with a dimension of nx1 to a vector and removing a row or column from a matrix which both require an element by element approach. Although these problems did not prevent the implementation of AUDI, it is clear that LabVIEW is not the ideal software for operations with complicated matrix operations.

6.3.5 Data Logging

Data logging utilities in LabVIEW were found to be both convenient and flexible. Data can be written to an ASCII file or to a spreadsheet file at each time interval or at the end of the run. Only one simple function was required to write the input-output data, controller constants and estimation parameters to a file.

6.3.6 Graphical User Interface

The front panel serves as the graphical users interface for LabVIEW programs. A wide assortment of front panel controls and indicators are available to interactively exchange information between the user and the block diagram. Front panel object attributes such as size, color, shape, decoration, data type etc. are easily modified to construct a custom interface. An example of such an interface used for the adaptive predictive PID controller described earlier is shown in Figure 6.4. The left portion of the front panel contains four clusters of parameters



Setpoint 35.00 Interactive Adaptive

Level sp, % 50.00

Drain valve, % 100.00

HW valve, % 6.00

Ts 4

Figure 6.4: Front panel of adaptive predictive PID for control of a steam heated stirred tank heater.

which are used to change the controller constants, setpoint, plant model, as well as process disturbance parameters such as level and drain valve signal interactively. Each of these clusters are in turn connected to the block diagram objects as shown in Figure 6.3. Trend charts on the right portion of Figure 6.4, used to display the setpoint, the plant output and the plant input, can be rescaled interactively simply by clicking on the numerical value and typing in the desired value. Only a few hours were required to construct this graphical interface which indicates that LabVIEW has very powerful and convenient graphical utilities suitable for control applications.

6.3.7 Data Acquisition

Three methods of data acquisition are possible with LabVIEW. First, is communication with instruments that use the IEEE-488 protocol, second, is specialized data acquisition cards that plug into the computer chassis and third is serial communications with an I/O subsystem. Since only the latter two methods were used in the experiments discussed in this thesis, this will form the focus of the following remarks.

The LabPC+ data acquisition card in conjunction with the LabVIEW DAC library was found to be very simple and effective when used for the light bulb control experiment presented in Chapter 2. Only one high level function is required for a read or write operation while numerous low level functions exist for more specialized data acquisition procedures.

Serial communications at 9600 BAUD to an existing OPTO 22 (Anon, 1990) I/O subsystem provided the data acquisition for the stirred tank heater experiments presented in Chapters 4 and 5. Some familiarity with the operation of OPTO 22 was required to construct the stirred tank control programs although this was a relatively minor issue. The OPTO 22 library within LabVIEW provided a powerful but simple suite of functions which required little effort for the construction of the stirred tank heater control applications. Because Windows is not a real time operating system, LabVIEW programs running under Windows can not truly be classified as real time applications. However, the actual execution of all LabVIEW programs was flawless for all experiments conducted in this thesis. It is concluded that data acquisition facilities within LabVIEW are ideal for experimental process control applications.

6.3.8 Program Execution and Debugging

Prior to running, LabVIEW programs are automatically compiled to machine code which results in fast program execution. However, it was observed that a significant decrease in execution speed occurs when the front panel contains waveform charts with the strip chart update mode and a data length that exceeds the x axis limit. It is therefore recommended that the sweep chart update mode be used for execution of time critical applications (e.g. experimental applications with a short sampling time).

Because of the visual nature of LabVIEW programs, debugging takes on a visual approach whereby step by step execution of the block diagram can be visually observed and probed. These debugging features were effective and simple to use.

6.4 Recommendations for Future Development

One of the most important issues regarding the future use of LabVIEW for experimental implementation of advanced process control and identification is integration with Matlab. Process control engineers and researchers have used Matlab extensively for many years to test and evaluate control and identification algorithms. Furthermore, many recent control and identification textbooks utilize Matlab (Franklin *et al.*, 1990; Ljung, 1987; Strum and Kirk, 1994; Leonard and Levine, 1992). It is clear from the discussions in section 6.3 that LabVIEW has a very powerful and flexible graphical users interface and excellent data acquisition utilities. With these comments in mind, the following recommendations are made regarding the future use of LabVIEW for experimental process control.

1. Control functions that currently exist as C source code programs can be compiled and used as subfunctions in LabVIEW programs with the use of the code interface utility in the LabVIEW advanced analysis library. In addition, it is possible to translate Matlab functions to C source code which can also be compiled and used as LabVIEW subfunctions to build experimental applications. This combines the most powerful features of LabVIEW with the most powerful features of Matlab and also allows existing Matlab code reuse. Although not tested thus far at the University of Alberta, it is believed these methods will be key in the future use of LabVIEW for experimental process control.
2. In Matlab version 4.2c and LabVIEW version 3.0.1, dynamic data exchange (DDE) is permitted in the Windows environment. This method remains to be evaluated at the University of Alberta. One disadvantage of this approach is that numerically intensive Matlab programs such as identification algorithms will require excessive computational resources because Matlab programs are executed in an interpretive manner.
3. Use LabVIEW exclusively (i.e. without Matlab or C subprograms) but take note of the comments in section 6.3. If a significant amount of time and effort has not been spent developing Matlab programs and if the programmer is sufficiently proficient in writing LabVIEW programs, this may be the best alternative.

6.5 Conclusions

- The LabVIEW learning curve was observed to be steep by undergraduate students and the author primarily because the programming methodologies differ significantly from text based programming languages.
- The graphical users interface and data acquisition utilities in LabVIEW for implementation of adaptive LRPC are excellent although complicated matrix operations were found to be excessively tedious.
- Combining LabVIEW and Matlab through the C code facility or through DDE may prove to be an ideal tool for experimental implementation of process control.

References

Anon, *OPTOMUX B1 and B2 Digital and Analog Brain Boards Operations Manual*, OPTO 22, Huntington Beach, CA, 1990.

Anon, *LabVIEW® for Windows User Manual*, National Instruments Corporation, Austin, TX, 1993.

Anon, *Matlab User Manual*, The MathWorks, Inc., Natick, Mass., 1994.

Franklin, G.F., J.D. Powell and M.L. Workman, *Digital Control of Dynamic Systems*, 2nd Ed., Addison-Wesley, Reading, MA, 1990.

Leonard, N.E. and W.S. Levine, *Using MATLAB to Analyze and Design Control Systems*, Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1992.

Ljung, L., *System Identification Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

Strum, R.D. and D.E. Kirk, *Contemporary Linear Systems using MATLAB*, PWS Publishing Co., Boston, Mass, 1994.

Chapter 7

Industrial Batch Identification Study

Batch identification of industrial input-output data using the Matlab® System Identification Toolbox and batch least squares based on an augmented upper diagonal factorization method are presented in this chapter. Model types suitable for use with generalized predictive control are evaluated using time and frequency domain analysis. ARX and ARMAX models were found to give a good fit to the plant dynamics which was validated by two independent data sets.

7.1 Introduction

With the recent increased use of model based control and on-line optimization in industry, the demand for simple but effective batch identification techniques has never been greater. The trend toward advanced control and process optimization will continue in the future as industrial corporations compete for a larger share of the global market. A popular vehicle for implementing advanced control is through model-based predictive control. It is well understood that such model-based advanced control techniques are effective only if the plant model is a reasonable approximation to the plant dynamics. It is also imperative to the success of advanced control that simple standardized identification techniques are available so that plant personnel can periodically validate and update plant models. The modelling process involves selecting the model type and the model structure in addition to determining the parameters of the model. The model type is normally chosen to accommodate the control application. For example, generalized predictive control (GPC) requires a linear time series (ARIMAX) type of model. The model structure, which includes the model order, time delay and noise model type, must be chosen carefully so that the most significant characteristics of the process are captured in the model. Lastly, the parameters of the model must be chosen to provide an optimal compromise between high frequency and steady state dynamics. The published literature is rich in the area of identification with thousands of reported applications and several textbooks.

The focus of this chapter is an identification case study using industrial open loop input-output data and standard analytical software. Two batch least squares techniques are considered in this study — both of which are standard available Matlab applications. The first is the Matlab System Identification Toolbox by Ljung (1992) which is available from The Mathworks, Inc. The second is the augmented upper diagonal identification (AUDI) method developed by Niu (1994) which is publicly available from the ftp site: *ftp.mathworks.com* in the directory: */pub/contrib/systemid/audidemo*. A linear time series type of model is selected to be compatible with GPC although the techniques presented in this chapter also apply to other model and control techniques. Although practical considerations such as time varying and nonlinear plant dynamics are important, it is assumed for the purposes of this study that the dynamics are relatively linear and time invariant.

7.2 Batch Least Squares Identification Techniques

Batch least squares (BLS) techniques are used by many disciplines to generate empirical models from input-output data. As far back as 1809 (Gauss, 1809), BLS was used for the development of models for the prediction of planetary orbits. Today, BLS and variants of BLS remain the most popular estimation techniques because of their simplicity and effectiveness. The intent of this section is to give a brief overview of estimating a simple time series model with batch least squares techniques.

7.2.1 Process Models

A linear time series model representation for the plant shown in Figure 7.1 can be expressed as

$$A\left(q^{-1}\right)y(t)=B\left(q^{-1}\right)u(t-1)+x(t) \tag{7.2.1}$$

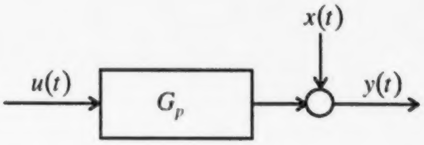


Figure 7.1: General process representation.

where $u(t)$ represents the process input, $x(t)$ is the disturbance, $y(t)$ is the noise corrupted output and

$$A = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{nA} q^{-nA} \qquad\qquad B = q^{-d}\left(b_1 + b_2 q^{-1} + \cdots + b_{nB} q^{-nB}\right)$$

where nA is the order of the plant denominator, nB is the order of the plant numerator and d is the time delay. The plant dynamic model, G_p, is then given by $\frac{B}{A}$ while the disturbance term, $x(t)$, can be expressed in several formulations. The disturbance models that apply to the GPC control law are given by the following expressions

$$\begin{array}{ll} \text{ARX} & x(t) = \xi(t) \\ \text{ARIX} & x(t) = \frac{1}{\Delta}\xi(t) \\ \text{ARMAX} & x(t) = C\left(q^{-1}\right)\xi(t) \\ \text{ARIMAX} & x(t) = \dfrac{C\left(q^{-1}\right)}{\Delta}\xi(t) \end{array} \qquad (7.2.2)$$

where $\xi(t)$ is a zero mean random sequence. The abbreviations in (7.2.2) apply when combined with (7.2.1) which are: auto-regressive (AR) applies to A and B; integrated (I) applies to $\frac{1}{\Delta}$; moving average (MA) applies to C; with exogenous input (X) applies to u. Rearrangement of (7.2.1) gives the following

$$y(t) = -a_1 y(t-1) - \cdots - a_{nA} y(t - nA) + b_1 u(t - d - 1) + \cdots + b_{nB} u(t - nB - d - 1) + x(t) \quad (7.2.3)$$

Such a model structure is also known as the equation error (EE) model. The main characteristic of the EE model structure being that the denominator dynamics due to the A polynomial are common to both the input and the disturbance.

7.2.2 Batch Least Squares

The ARX representation of (7.2.3) can be expressed as a product of a parameter vector and an input-output data vector given by

$$y(t)=\phi_{BLS}^{T}(t)\hat{\theta}_{BLS}+\xi(t) \tag{7.2.4}$$

where the parameter vector is

$$\hat{\theta}_{BLS}=\begin{bmatrix}a_1 & a_2 & \cdots & a_{nA} & b_1 & b_2 & \cdots & b_{nB}\end{bmatrix}^T \tag{7.2.5}$$

and the data vector is

$$\phi_{BLS}(t)=\begin{bmatrix}-y(t-1) & -y(t-2) & \cdots & -y(t-nA) & u(t-d-1) & u(t-d-2) & \cdots & u(t-d-nB)\end{bmatrix}^T \tag{7.2.6}$$

The objective of BLS in this case is to determine the parameter vector (7.2.5) that minimizes the sum of the single step ahead prediction errors expressed as

$$J=\sum_{j=t_1}^{t_2}\left[y(j)-\phi_{BLS}^{T}(j)\hat{\theta}_{BLS}\right]^2 \tag{7.2.7}$$

The parameter vector which minimizes (7.2.7) from the input-output data in the series $t=t_1\cdots t_2$ is given by

$$\hat{\theta}_{BLS}=\left[\sum_{j=t_1}^{t_2}\phi_{BLS}(j)\phi_{BLS}^{T}(j)\right]^{-1}\sum_{j=t_1}^{t_2}\phi_{BLS}(j)y(j) \tag{7.2.8}$$

which is essentially a multiple linear regression. A proof of (7.2.8) can be found in almost every textbook on process identification or statistics (e.g. Ljung, 1987). It should be noted that the BLS solution in (7.2.8) can be applied to any of the model types in (7.2.2).

7.2.3 Augmented Upper Diagonal Identification

The AUDI method for process identification is a variant of least squares. As such, the objective of batch AUDI is exactly the same as BLS. The major differences between AUDI and BLS for batch identification are the formulation of the augmented parameter and data vectors and the numerical solution of the parameter vector. This section is intended to give a very brief account of the AUDI method for estimating ARX models. ARIX models can be estimated by differencing the input-output data. For a thorough treatment of the topic and detailed derivation of the AUDI algorithm, see Niu *et al.* (1992) or Niu (1994).

The augmented parameter and data vectors are defined by defining the elements of the parameter and data vectors as follows

$$\phi(t)=\left[-y(t-n)\quad u(t-n-d)\quad \ldots\quad -y(t-1)\quad u(t-1-d)\quad -y(t)\right]^T \tag{7.2.9}$$

$$\hat{\theta}(t)=\left[a_n\quad b_n\quad \ldots\quad a_1\quad b_1\quad 1\right]^T \tag{7.2.10}$$

The ARX model based on the augmented vectors can then be expressed as

$$0=\phi^T(t)\hat{\theta}(t)+\xi(t) \tag{7.2.11}$$

A special covariance matrix denoted as the augmented information matrix (AIM) is defined as

$$C(t)=\left[\sum_{j=t_1}^{t_2}\phi(j)\phi^T(j)\right]^{-1} \tag{7.2.12}$$

The AIM is then factored into the following diagonal and upper diagonal matrices

$$C(t)=U(t)D(t)U^T(t) \tag{7.2.13}$$

where the U and D matrices are defined by

$$U(t)=\begin{bmatrix} 1 & \hat{\alpha}_1^1 & \hat{\theta}_1^1 & \cdots & \hat{\alpha}_1^n & \hat{\theta}_1^n \\ & 1 & \hat{\theta}_2^1 & \cdots & \hat{\alpha}_2^n & \hat{\theta}_2^n \\ & & 1 & \cdots & \hat{\alpha}_3^n & \hat{\theta}_3^n \\ & & & \ddots & \hat{\alpha}_4^n & \hat{\theta}_4^n \\ & & & & \vdots & \vdots \\ & 0 & & & 1 & \hat{\theta}_{2n}^n \\ & & & & & 1 \end{bmatrix} \qquad D^{-1}(t)=\begin{bmatrix} J^0 & & & & & \\ & L^0 & & & 0 & \\ & & J^1 & & & \\ & & & \ddots & & \\ & 0 & & & L^{n-1} & \\ & & & & & J^n \end{bmatrix} \tag{7.2.14}$$

where $\hat{\theta}$ is the model given by (7.2.10), $\hat{\alpha}$ contains the parameters of a different model, J is the loss function of the $\hat{\theta}$ model given by (7.2.7) and L is the loss function of the $\hat{\alpha}$ model. The superscripts and subscripts in (7.2.14) correspond to the model order and element of $\hat{\theta}$ or $\hat{\alpha}$, respectively. For about the same effort as the BLS solution, AUDI provides estimates and the loss functions of all ARX models from order one to n. In the AUDI algorithms available at the previously mentioned Mathworks ftp site, either LU, QR or Cholsky decomposition of $C^{-1}(t)$ can be selected by the user. Therefore, AUDI does not involve computation of a matrix inverse which has significant numerical advantages compared to BLS for low plant input excitation.

7.2.4 The Matlab System Identification Toolbox

The functions contained in the Matlab System Identification Toolbox (denoted as ID toolbox) provide the means to estimate time series and state space type of models from input-output data. Frequency and time domain techniques are employed to validate and analyze the model. This section briefly discusses the techniques used in the estimation of ARX and ARMAX models by the ID toolbox in order to provide a basis for comparison with AUDI. As was the case in the AUDI method, integrating models can be estimated by differencing the input-output data. Readers interested in the specific use of the ID toolbox are urged to consult the user's guide (Ljung, 1993). Throughout the remainder of this chapter italicized commands represent functions in the ID toolbox.

The *arx* function in the ID toolbox is used to estimate the ARX model given by (7.2.3). For a time series of input-output data and model order *n*, (7.2.3) can be expressed as

$$\begin{bmatrix} \hat{y}(t_1) \\ \hat{y}(t_1+1) \\ \vdots \\ \hat{y}(t_2) \end{bmatrix} = \begin{bmatrix} \phi^T_{BLS}(t_1) \\ \phi^T_{BLS}(t_1+1) \\ \vdots \\ \phi^T_{BLS}(t_2) \end{bmatrix} \hat{\theta}_{BLS} \quad \text{or} \quad \mathbf{y}=\mathbf{\Phi}\hat{\theta}_{BLS} \tag{7.2.15}$$

The solution of (7.2.15) employed by the ID toolbox uses the Matlab function "\" for computing the solution of the parameter vector given by

$$\hat{\theta}_{BLS} = \mathbf{\Phi} \setminus \mathbf{y} \tag{7.2.16}$$

ARMAX models are estimated by using the *armax* command in the ID toolbox which employs a robustified Gauss-Newton algorithm to iterate the solutions of A, B and C. Iteration is required because a non-unity value of C requires the current noise input, $\xi(t)$, which is estimated by the residual, $\hat{\xi}(t) = y(t) - \hat{y}(t)$.

The solution matrix returned by the *arx* and *armax* functions (denoted as *theta*) contains the loss function, model uncertainty and the model estimate in addition to other relevant information about the identification result. The element *theta*(1,1) contains the loss function of the model which differs slightly from the definition in (7.2.7) given by

$$J = \frac{1}{N - \dim} \sum_{j=t_1}^{t_2} \left[y(j) - \hat{y}(j) \right]^2 \tag{7.2.17}$$

where N is the data length (or t_2-t_1+1) and dim is max(nA, nB+d+1)-1. This is a very useful loss function definition because it also approximates the noise variance for a large N.

7.3 Identification Case Study

The objective of this case study is to estimate a dynamic model of an industrial process that is suitable for use with a GPC controller. The process consists of a highly exothermic catalytic reactor in the fertilizer plant described in Chapter 2. In relation to the general representation in Figure 7.1, the plant input is the reactant flow rate and the plant output is the catalyst temperature. Three open loop runs were performed on the reactor in which step changes were made to the reactant flow rate while the catalyst temperature was recorded at five second intervals which is displayed in Figure 7.2. The approach taken in this study is to use one data set for identification (denoted as ID data set) and the remaining two data sets for validation (denoted as validation set 1 and validation set 2) of the model. It is essential for reliable identification that the sample mean and linear trends are removed from the input-output data. Therefore, the Matlab ID Toolbox command *dtrend(z,1)* was used to condition the industrial data prior to all estimation results presented in this chapter.

7.3.1 ARX Model Estimation

7.3.1.1 Selection of Time Delay and Model Order

The phase contribution of the time delay can be a dominant factor in the closed loop stability of model based control. Significant model plant mismatch can occur when the time delay is either under or over estimated. In addition, over estimating the model order can lead to numerical instability of the plant model estimate. The selection of time delay and model order is the first

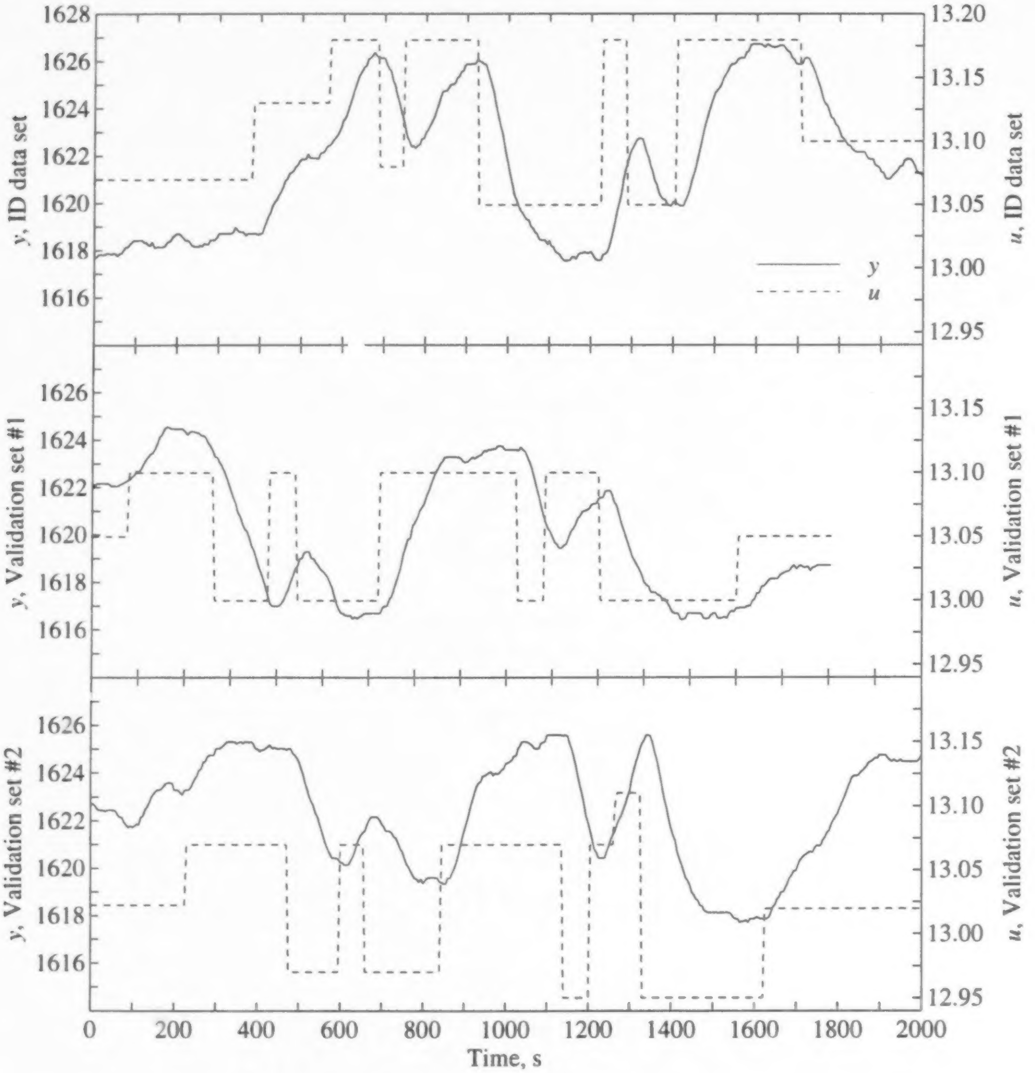


Figure 7.2: Open loop industrial input-output data.

procedure in the identification process because these are integral components of the model structure and they affect the structure of the ϕ_i vector. Although the actual time delay of the process is not normally related to the actual order of the plant dynamics, the estimated time delay is strongly related to the chosen order of the plant model (e.g. the time delay that gives the best model-plant fit for a first order model is not the same time delay that gives the best fit for a higher order model). It is therefore important that the model order and time delay are chosen in a simultaneous fashion. A computationally intensive but effective method of simultaneously choosing time delay and model order consists of estimating all of the models for a wide range of time delay and model order. The ID toolbox function *arxstruc* computes the loss functions according to (7.2.17) for any given range of ARX model structures. As shown in the top plot of Figure 7.3, the optimal time delay increases with decreasing model order. Cross-validation is performed (an option of *arxstruc*) by selecting one data set for identification and a different data set for computation of the loss function. Comparison of the top plot with the bottom two plots of Figure 7.3 indicate that the cross-validation loss functions are consistent with the ID data set loss functions. It is clear that no single estimate of the time delay is optimal for all model orders. The approach taken in this study is to base the selection of model order on the optimal time delay estimate of each model order rather than a single time delay for all model orders.

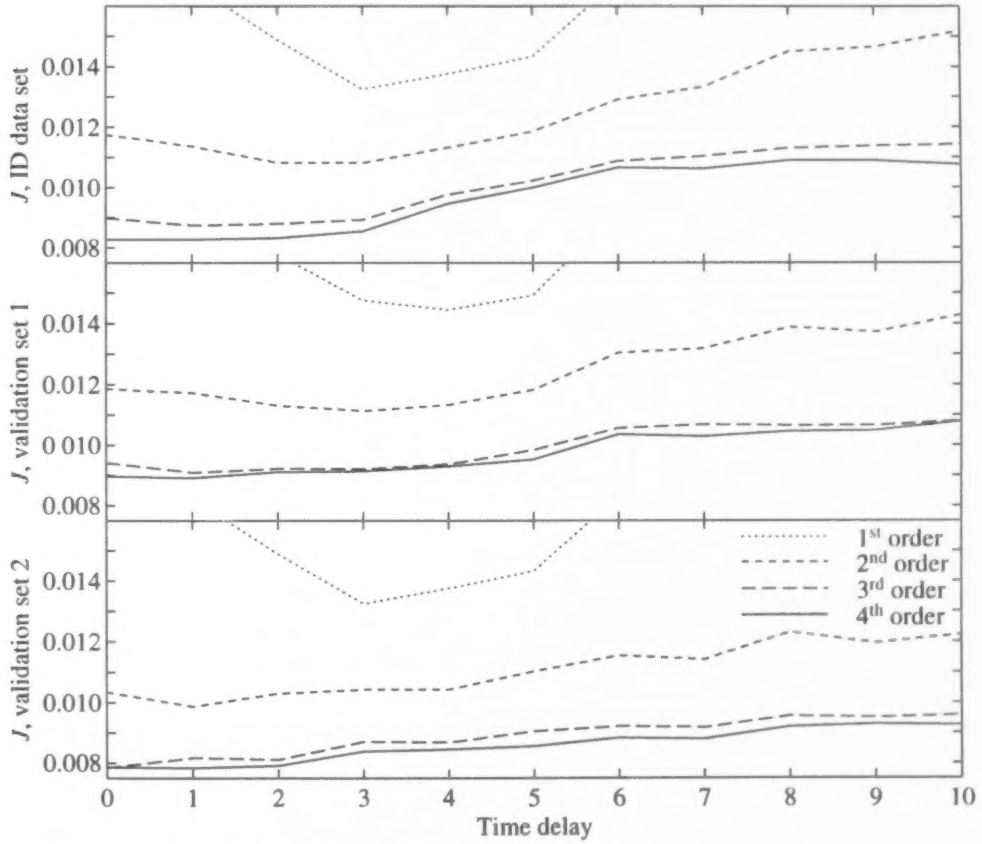


Figure 7.3: ARX model loss function plots with respect to time delay using the ID toolbox.

Cross-correlation analysis can also give an indication of time delay for input-output data. The ID toolbox function *cra* computes the cross-correlation between y and u following prefiltering the data. The plot in Figure 7.4 shows the first 11 estimated impulse response coefficients computed by *cra* in addition to the 99 % confidence level. A delay of between 3 and

4 sample intervals is indicated although the confidence level is very high. Significantly more data than is available in these industrial open loop data sets are required to obtain meaningful time delay estimation results from cross-correlation analysis.

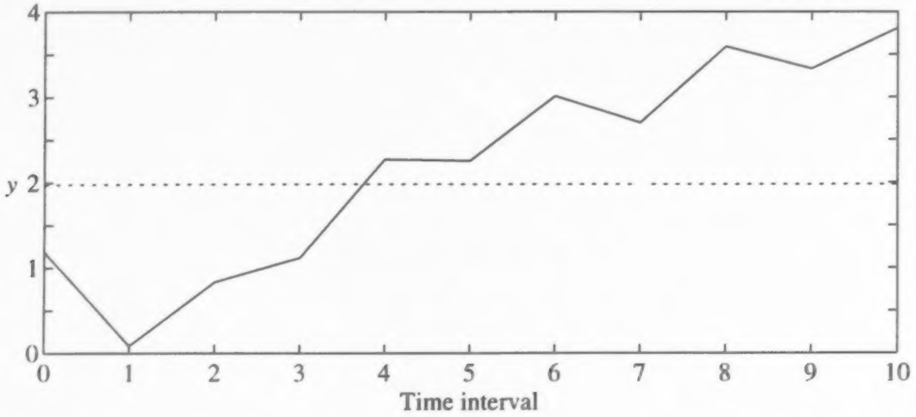


Figure 7.4: Impulse response estimate of the ID data set using the ID toolbox function *cra*.

The loss functions corresponding to the optimal time delay estimate at each model order are presented in Figure 7.5 for the ID data set as well as the validation sets. Loss functions computed by AUDI in Figure 7.5 are multiplied by 1/(N-dim) to be consistent with the loss function definition in (7.2.17). Although AUDI and the ID toolbox are both based on a least squares algorithm, the solutions are not exactly the same presumably because of the differences in the numerical solution (i.e. pseudoinverse vs. the UDU^T factorization) and data conditioning in the ID toolbox. There is a significant improvement in loss function for increasing the model order up to three but further increases result in a negligible improvement as shown in Figure 7.5. High model orders increase the computational load of the application and may result in numerical instability. Intuitively, the best choice on model order from the loss function plot in Figure 7.5 is three because higher orders give minimal improvements. The model order selection process can be formalized by using an objective function that penalizes high dimensionality. The following expression is one such objective function used by the ID toolbox function *selstruc*.

$$J_{\text{mod}} = J\left(1 + cN_p/N\right) \tag{7.3.1}$$

where c is the penalty applied to the dimension and N_p is the sum of the order of A and B (i.e. for a third order model $N_p = 6$). The Akaike information theoretic criterion (AIC) as used by Ljung (1987) is described by (7.3.1) with $c = 2$. The "optimal" model order is then given by the model

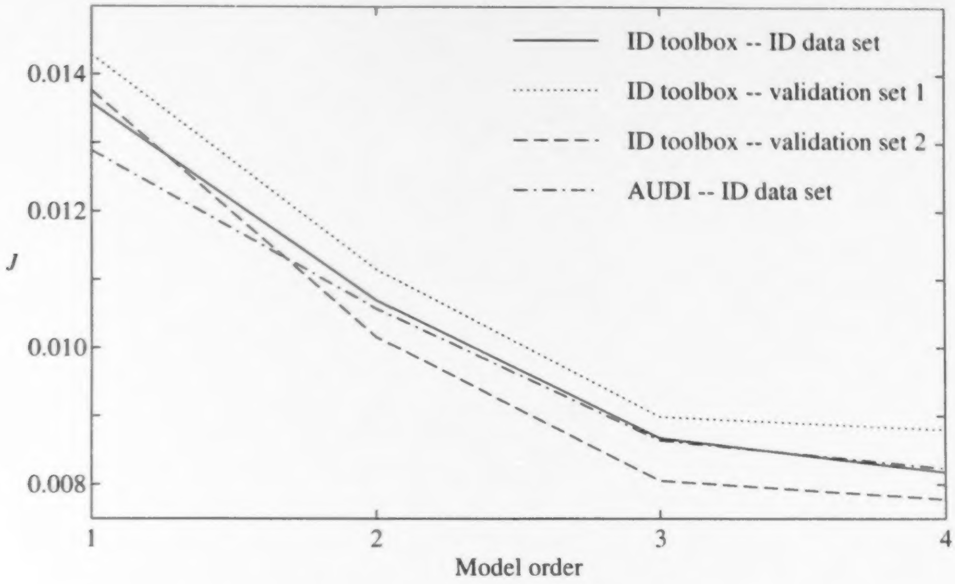


Figure 7.5: Loss function plot with respect to model order for ARX models.

that minimizes (7.3.1). Model structure selection based on AIC (i.e. $c = 2$) and $c = 5$ in (7.3.1) using the ID toolbox function *selstruc* is presented in Table 7.1. Based on the results of *selstruc*, the optimal model order and time delay are three and one, respectively.

Table 7.1: Optimal ARX model order and time delay using the ID toolbox function *selstruc*.

	ID data set		Validation set 1		Validation set 2	
	AIC ($c = 2$)	$c = 5$	AIC ($c = 2$)	$c = 5$	AIC ($c = 2$)	$c = 5$
Model order	4	3	4	3	3	3
Time delay	1	1	1	1	0	0

Model estimation of the detrended data based on the optimal ARX structure using AUDI and the ID toolbox is presented in Table 7.2. Coefficients of the A polynomial estimated by AUDI and the ID toolbox show close agreement, but the B polynomial coefficients differ significantly. Similarly, the poles of the AUDI solution match closely with the ID toolbox solution while the zeros are significantly different. In addition, the model generated from the ID toolbox is non-minimum phase while the AUDI solution is minimum phase. It is therefore expected that the high frequency dynamics will differ somewhat while low frequency dynamics will be very similar.

Table 7.2: Estimated ARX model parameters using AUDI and the ID toolbox.

ID Method	Plant Model $G_p = \frac{q^{-d}B}{A}$	Zeros	Poles	Gain
AUDI	$\frac{q^{-1}\left(0.9718 + 0.8805q^{-1} + 0.3510q^{-2}\right)}{1 - 1.1168q^{-1} - 0.2198q^{-2} + 0.3697q^{-3}}$	-0.4530±j0.3949	0.9163 0.7434 -0.5428	66.5
ID toolbox	$\frac{q^{-1}\left(0.5606 + 0.4092q^{-1} + 1.1516q^{-2}\right)}{1 - 1.1329q^{-1} - 0.2066q^{-2} + 0.3709q^{-3}}$	-0.3650±j1.3861	0.9172 0.7529 -0.5372	67.4

7.3.1.2 Time Domain Validation

Cross-validation of dynamic models is an essential component of the identification process. This ensures that the model is a good fit to the plant dynamics and not just to a specific set of input-output data. Two time domain methods are employed to validate the models in Table 7.2. The first is the loss function (7.2.17) which is computed independent of the results in section 7.3.2.1 to ensure consistency between the ID toolbox and AUDI. The second method is the mean square fit (computed by the ID toolbox function *compare*) defined by

$$MSF = \frac{\sqrt{\sum \left|y - y_{sim}\right|^2}}{\sqrt{N}} \tag{7.3.2}$$

where y_{sim} is the simulated response with the plant input data. Table 7.3 shows the loss functions and mean square fit for the ARX models in Table 7.2 applied to all input-output data sets. As expected, the *MSF* is best for the ID data set although the fit is acceptable for the validation data sets as well. The sum of *J* and the sum of *MSF* for all data sets matches very closely for the models generated by AUDI and the ID toolbox. The predicted plant output trajectory using the ARX models in Table 7.2 (without actual plant output feedback) in response to the plant input

Table 7.3: Time domain validation of ARX model estimates.

	ID data set		Validation set 1		Validation set 2		combined	
	J	*MSF*	*J*	*MSF*	*J*	*MSF*	ΣJ	ΣMSF
ID toolbox	0.0087	0.5390	0.0090	0.7646	0.0080	0.7819	0.0257	2.0855
AUDI	0.0089	0.5262	0.0094	0.7655	0.0082	0.7903	0.0265	2.0819

data is plotted with the actual plant output data in Figure 7.8. The models generated by AUDI and the ID toolbox produce simulations that are almost indistinguishable. The observed fit between the simulations and the ID data is excellent. Although the observed fit between the simulations and the validation data is not as good as the ID data set, it is still very acceptable. It is not surprising that the model plant mismatch is the smallest for the ID data set because the model was estimated from the same data.

7.3.2 ARIX Model Estimation

An ARIX model was estimated by differencing the input-output data and following the same procedure as in section 7.3.1. To make a long story short, the ARIX model can not adequately represent the dynamics of the industrial reactor. As shown in Figure 7.6, the simulated response of the ARIX models is a very poor fit to the plant output data. Further investigation is not required to conclude that the reactor dynamics are not suited to an integrating noise model.

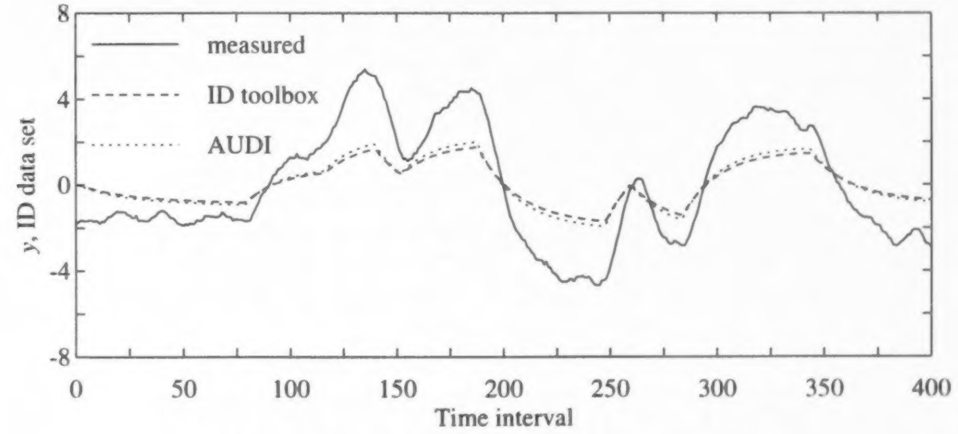


Figure 7.6: Simulated response of the ARIX model.

7.3.3 ARMAX Model Estimation

Identification of ARMAX models requires iteration as mentioned in section 7.2. Since the AUDI routines available are not formulated for batch identification of ARMAX models, the identification results presented in this section are performed only by the ID toolbox. The only difference in the analysis of this section compared to section 7.3.1 is that the analogous function to *arxstruc* does not exist for ARMAX models. Instead, the ARMAX model for each model structure was identified using the *armax* function and a separate function was written to compute

J_{mod} for selection of time delay and model order.

7.3.3.1 Selection of Model Order and Time Delay

Selection of the optimal structure for models of the ARMAX type follows the same procedure as in section 7.3.1. Although higher order C polynomials were evaluated in the ARMAX model estimation, a first order C resulted in the best fit and was used in all subsequent analysis. The loss functions corresponding to the optimal time delay estimate at each model order are presented in Figure 7.7 for the ID data set and the validation sets. Significant improvements in loss functions are shown by increasing the model order from one to two but further increases in model order results in diminishing improvements. The intuitive choice of model order is either two or three depending on the constraints of the application. Selection of an ARMAX model structure based on the AIC and $c = 10$ in (7.3.1) criteria is presented in Table 7.4. Additional improvements in loss function by increasing the model order to three from two exceed the penalty on the higher model dimension even when a strong weight is applied to the selection criterion. The optimal time delay and model order for the ARMAX model estimation is therefore chosen to be one and three, respectively.

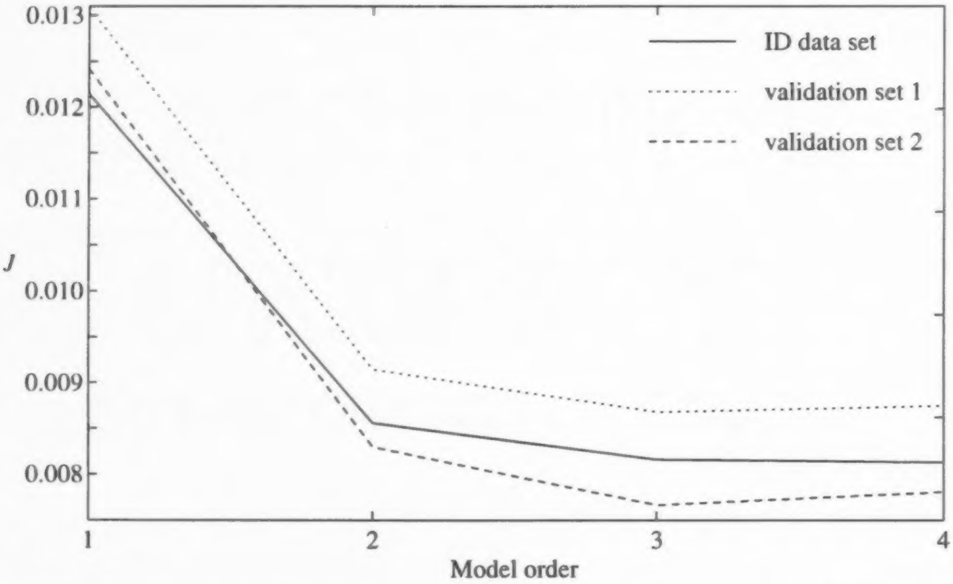


Figure 7.7: Loss function with respect to model order for the ARMAX model with a first order C polynomial.

Table 7.4: Optimal ARMAX model order and time delay using the ID toolbox and (7.3.1).

	ID data set		Validation set 1		Validation set 2	
	AIC ($c = 2$)	$c = 10$	AIC ($c = 2$)	$c = 10$	AIC ($c = 2$)	$c = 10$
Model order	3	3	3	2	3	3
Time delay	1	1	1	1	0	0

Model estimation based on the optimal ARMAX structure using the ID toolbox is presented in Table 7.5. Coefficients of the *A* polynomial and the poles are similar to those estimated in Table 7.2 while the *B* polynomial coefficients are somewhat different than in Table 7.2. Furthermore, the zeros indicate that the model exhibits non-minimum phase behavior while the gain is slightly lower than the gain estimated by the ARX models. The *C* polynomial estimate indicates that the plant noise is not strongly correlated.

Table 7.5: Estimated ARMAX model parameters using the ID toolbox.

Plant Model $G_p = \frac{q^{-d}B}{A}$	Noise Model $x(t) = \frac{C}{A}\xi(t)$	Zeros	Poles	Gain
$\frac{q^{-1}\left(0.6336 + 0.1184q^{-1} + 0.6590q^{-2}\right)}{1 - 1.4569q^{-1} + 0.2456q^{-2} + 0.2332q^{-3}}$	$\frac{1 - 0.4414q^{-1}}{A}$	-0.0934±j1.0155	0.8789±j0.0468 -0.3010	64.4

ARMAX model: $Ay=Bu+C\xi$ (equation error structure)

7.3.3.2 Time Domain Model Validation

The time domain performance of the ARMAX model presented in Table 7.6 shows a slight improvement in every case compared to the ARX model performance in Table 7.3. The *MSF* of the simulated model also follows the same pattern as the ARX model which is best for the ID data set and acceptable for the validation sets. The simulated response of the ARMAX model in Table 7.5 to the plant input data is plotted with the actual plant output data in Figure 7.8. The ARMAX simulated response is an excellent fit to the plant data although it is barely distinguish-

Table 7.6: Time domain validation of the ARMAX model.

ID data set		Validation set 1		Validation set 2		combined	
J	MSF	J	MSF	J	MSF	ΣJ	ΣMSF
0.0082	0.4990	0.0087	0.7510	0.0079	0.7741	0.0247	2.0241

able from the ARX model simulations. Unit step responses of all models estimated in this chapter are displayed in Figure 7.9. As can be seen, the response of the ARX and ARMAX models are very similar while the ARIX model differs significantly. The cumulative sum of the Matlab *cra* impulse response as shown in Figure 7.4 is also plotted in Figure 7.9. Good transient response agreement between the *cra* and the ARX and ARMAX models is shown although a steady state of the cumulative sum of the *cra* model cannot be obtained because the impulse coefficients converge close to but not exactly zero.

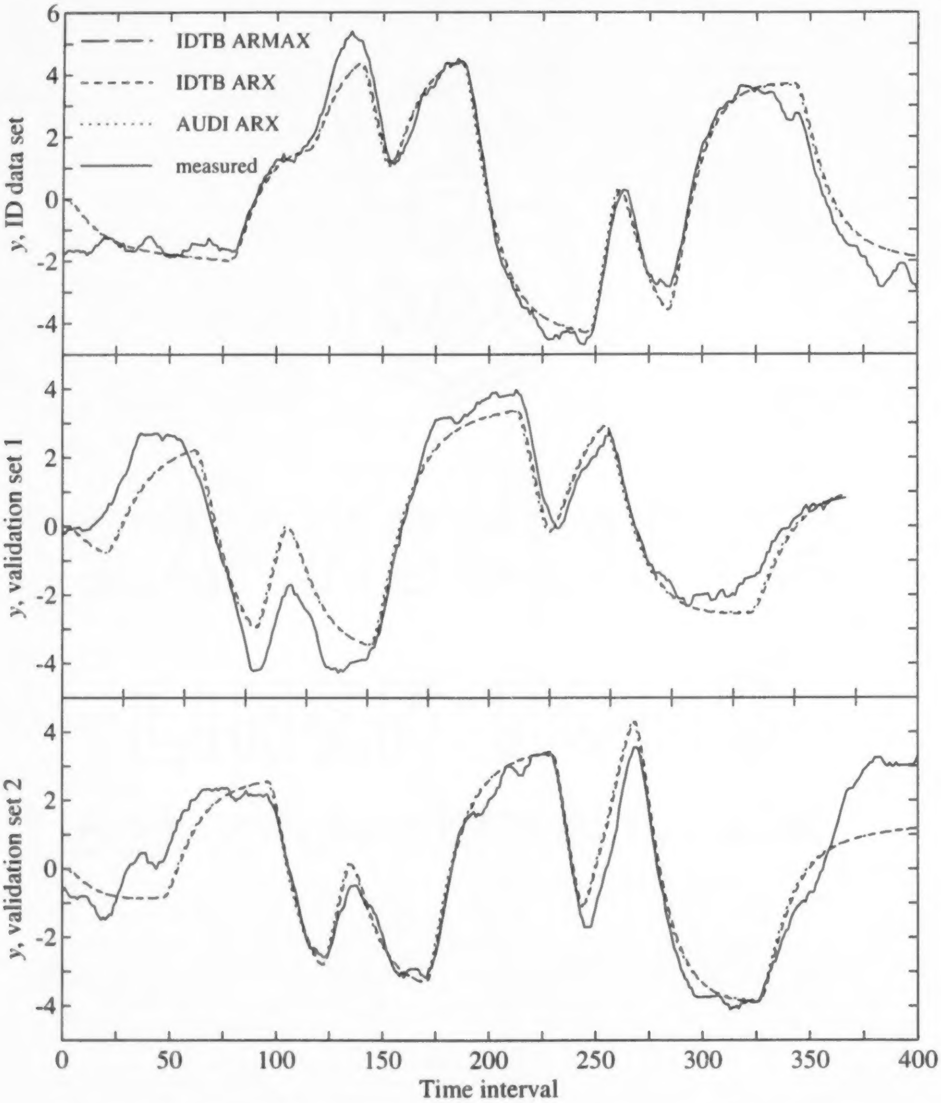


Figure 7.8: Simulated response of ARX and ARMAX models to all data sets.

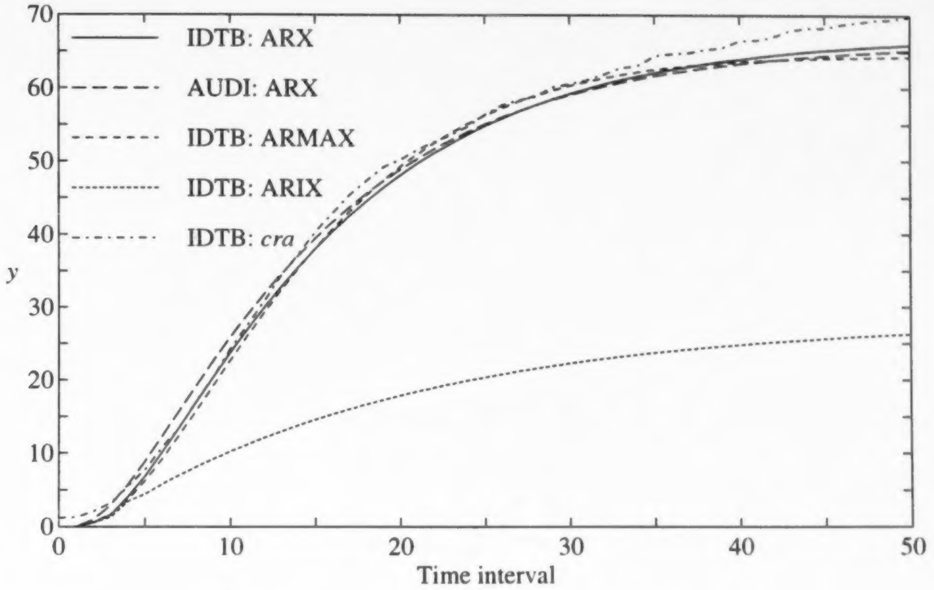


Figure 7.9: Unit step responses of plant models.

7.3.4 Analysis of Residuals

A common technique used to validate the ability of a model to fit the dynamics of the plant and the noise is to perform an auto-correlation analysis on the residuals or prediction errors. If the noise model is a reasonable representation of the true noise dynamics, the residuals will not be auto-correlated. Conversely, strong auto-correlation of the residuals indicates that the noise model is inappropriate. The auto-correlation of the residuals along with the 99 % confidence interval (from the ID toolbox function *resid*) is plotted for the ID data set in Figure 7.10. As expected, the residuals from the ARIX model are strongly correlated which indicates that ARIX is a poor model choice. It is also not surprising that the residuals from the ARMAX model are virtually uncorrelated because the noise model was specifically estimated along with the plant model parameters. Although the ARX models show a small amount of auto-correlation, there is only one violation of the confidence interval for each ARX model as shown in Figure 7.10.

7.3.5 Frequency Domain Model Validation

Spectral analysis techniques such as fast Fourier transforms offer a convenient means to validate a model with input-output data in the frequency domain. Both low frequency and high frequency dynamics can be evaluated by computing the magnitude and phase of the models at various

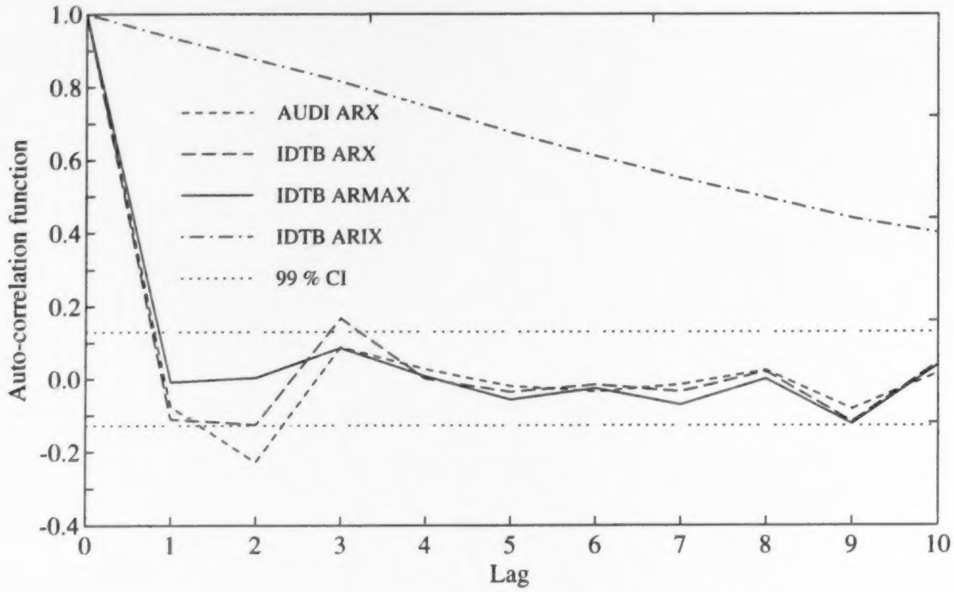


Figure 7.10: Auto-correlation of residuals for the ID data set.

frequencies. In Matlab this is executed with the function *bode* while the frequency response of the plant data is evaluated with the *spa* command. Standard Bode analysis methods can be used to evaluate the fit of the model to the plant spectrum. Figure 7.11 shows the frequency response of the plant models in Tables 7.2 and 7.5 and the frequency spectrum of the plant data. The magnitude plot shows good agreement between the models and the plant at low frequencies which means the steady state gain is estimated correctly. Attenuation at mid frequencies of the plant magnitude plot is also followed very closely by all of the models until about 0.06 Hz which indicates the dominant time constant of the models matches the plant dynamics. Noise is dominant at high frequencies which explains the significant deviation between the models and the plant spectrum for both magnitude and phase plots. Good agreement in phase plots between the plant and all models is shown for low and mid frequencies. All of the ARX and ARMAX models therefore match the low and mid frequency dynamics of the plant. In order to evaluate the high frequency fit, significantly more plant data and stronger filtering are required to increase the reliability of the plant spectrum.

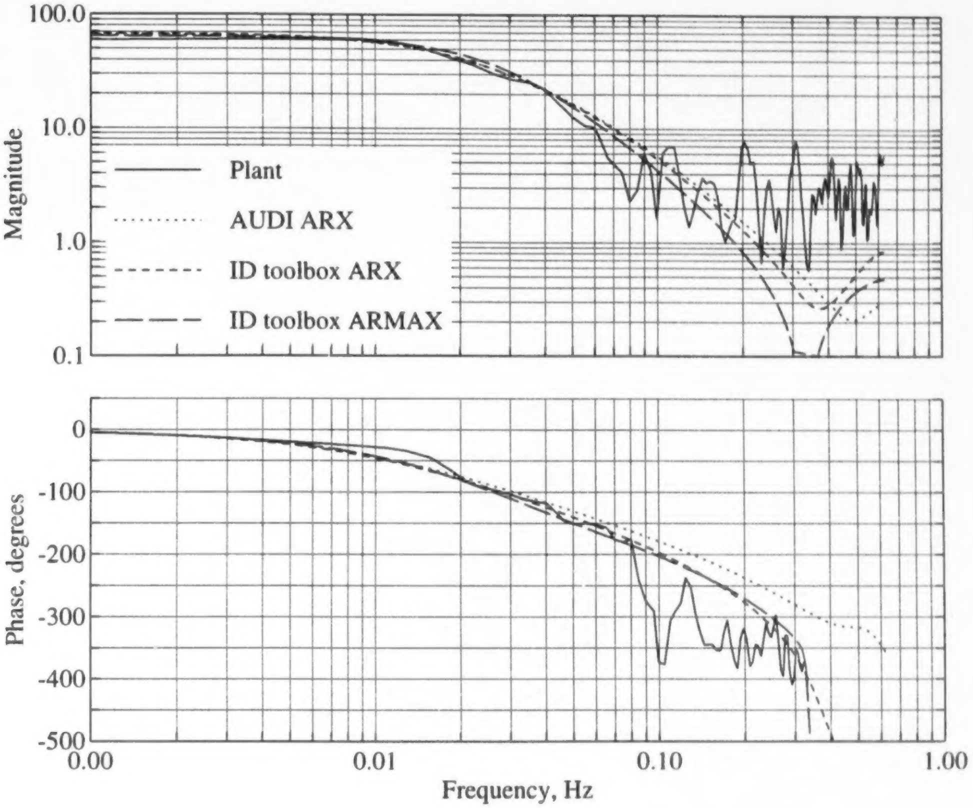


Figure 7.11: Frequency response of the plant data and models.

7.4 Conclusions

- ARX and ARMAX model structures are a good fit to the plant reactor input-output data which was validated in the time and frequency domains.
- Analysis of the residuals shows a weak correlation for ARX models which motivates the use of the ARMAX type model.
- Both AUDI and the System Identification Toolbox provided a convenient means for batch identification of plant models. The discrepancy in ARX models is due to numerical differences in the algorithms and the data conditioning in the ID toolbox. The System Identification Toolbox was found to be is a very convenient and powerful tool for the identification and validation plant models from input-output data.

References

Gauss, K.F., "Theoria motus corporum coelestium (1809) English translation: Theory of motion of the heavenly bodies," Dover, New York, 1963.

Ljung, L., *System Identification Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

Ljung, L., *System Identification Toolbox for use with Matlab*, The MathWorks, Inc., Natick, Mass., 1992.

Niu, S., D.G. Fisher and D. Xiao, "An augmented UD identification algorithm," *Int. J. Control*, Vol. 56, No. 1, pp. 193-211, 1992.

Niu, S., "Augmented UD identification for process control," *Ph.D. thesis*, University of Alberta, 1994.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

This thesis has addressed two main issues: first, the development of a model based PID control law that is equivalent to a long range predictive control law and second, the on-line estimation of a plant time delay from an extended numerator model. In addition, several supporting topics were investigated including an overall performance measure and an industrial batch identification case study. A summary of the main contributions is given as follows:

1. Predictive PID Control Law

A predictive PID control law results from equating the linear polynomials of GPC (Kwok and Shah, 1994) with the linear polynomials of an incremental form of a PID control law with proportional and derivative action removed from setpoint changes. The third term in the linear GPC control law constitutes the internal model (denoted as G_{MP}) for the predictive PID algorithm which exists only if there is a time delay or colored noise in the plant model. Therefore, plant time delays and stochastic disturbances are compensated for in an optimal fashion by G_{MP} which can also be interpreted as a multistep weighted predictor. First and second order plant models yield model based PI and PID controllers, respectively. Thus, the maximum plant model order is two although there is no restriction on the GPC controller constants or the noise model characteristics on which the predictive PID law is based. For the case where the *C* polynomial in GPC is non unity, there is a slight approximation required in the predictive PID servo term which will always result in a detuned servo response compared to GPC. An *ad-hoc* solution to increase the servo aggression of predictive PID for cases where servo response is too slow simply requires including proportional action for setpoint changes. Setting the finite prediction weight in GPC to zero results in a mean level formulation of predictive PID for which the PID controller constants can be solved explicitly in terms of the plant model parameters. This formulation of the predictive PID controller is especially attractive because it is very simple and requires minimal effort to compute the PID controller constants. Two block diagram representations of predictive PID were presented to facilitate its implementation and interpret the model based structure. Simulations of predictive PID prove that it is equivalent to GPC for regulatory control while for servo control it is slightly more sluggish when colored noise is specified in the plant model.

2. Extended Numerator Rationalization for On-line Time Delay Estimation

A new method of on-line time delay estimation was presented based on the rationalization of the numerator coefficients of an extended numerator model. It was proved that for a first order ARX model with two numerator coefficients, the time delay can be represented by the relative weight of the second *B* coefficient. The main idea behind the ENR technique is to extend this concept to any number of *B* coefficients using a method of moments. It is recommended that a first order ARX model with enough numerator coefficients to cover the expected range of plant time delay and a fixed denominator coefficient be recursively updated using AUDI (Niu *et al.*, 1992). Fixing the denominator of the time delay model was shown to increase the sensitivity of the numerator following changes in the plant time delay. Recursive AUDI was chosen because the additional information contained in the *U* and *D* matrices proved to be key for the practical implementation of ENR. An example of ENR applied to a first order plant with open loop excitation and a time varying delay was shown to give accurate integer and fractional estimates of the time delay. It was demonstrated that an estimate of the ENR uncertainty can be calculated based on the propagation of variance of the estimated *B* coefficients.

Simulation results showed that the ENR technique is as good or better than the variable regression estimation method of Elnaggar *et al.* (1991) under closed loop servo and regulatory operation. Concluding remarks regarding the experimental evaluation of ENR in combination with an adaptive predictive PID control law follow in the next section.

3. Practical Evaluation of Predictive PID, Adaptive Predictive PID and ENR

Two potentially successful areas for the practical application of predictive PID controllers are the replacement of conventional PID in industrial control computers and adaptive or self-tuning PID when combined with an on-line identification algorithm such as AUDI. The main focus of all of the experimental tests performed on the stirred tank heater and the industrial implementation was regulatory control performance to be consistent with the primary objective of industrial process control.

Predictive PID was implemented using an industrial control computer for control of an industrial heat exchanger by specifying a user defined function for G_{MP} and removing proportional and derivative action from setpoint changes which are built in options. Only one hour was required to implement the predictive PID algorithm on the TDC2000 control computer not including the modelling and controller design. This represents a significant savings in implementation costs compared to any other advanced control scheme. In comparison with the existing PID controller, the performance of the predictive PID control algorithm was superior as indicated by the significantly lower variance of the controlled variable as well as the variance of the manipulated variable during regulatory control.

Extensive experimental testing of the predictive PID controller combined with AUDI for control of the discharge temperature of a stirred tank heater permitted a thorough evaluation of the algorithm's performance. Based on the findings of Shook *et al.* (1991), the input-output data were filtered using a bandpass filter prior to parameter estimation using AUDI to make the identification objectives compatible with the long range objectives of predictive PID. Varying degrees of disturbances to the discharge temperature were introduced some of which resulted in a significant change in the plant dynamics. Excellent regulatory control performance was demonstrated by the adaptive predictive PID controller during all of the runs. The improvement in performance of adaptive predictive PID compared to predictive PID increased with the magnitude of the change in plant dynamics. In response to a significant increase in the time delay, the gain and time constant of the estimated model were overestimated by a factor of six times which resulted in a detuned but stable performance. However, it is possible that this phenomenon is a case specific occurrence.

Adaptive predictive PID in combination with ENR proved to be effective for control of the stirred tank heater during upsets that caused the time delay to vary significantly. The ENR estimate of the time delay responded quickly and accurately to changes in the time delay which resulted in an improvement in performance compared with the adaptive predictive PID controller based on a fixed time delay model. During the period of increased time delay, the parameters of the plant model converged approximately to the correct values that were identified by open loop tests and this resulted in significantly improved performance during subsequent disturbances. It is concluded that adaptive predictive PID in combination with ENR for control of time delay varying processes is a very effective control scheme.

4. Robustness Margin and an Overall Performance Measure

A new scalar robustness margin based on the small gain theorem was proposed for closed loop feedback control systems subject to model plant mismatch. In addition, an overall performance measure which includes a traditional performance measure such as ISE subject to a penalty on the square of the incremental control action and a penalty on the inverse of the margin of robustness was introduced. Several examples were presented to demonstrate use of these measures for assessment of control performance while varying two control or plant parameters. The resulting 3-dimensional surface provides a graphical view of control performance with respect to tuning parameters and model plant mismatch.

5. LabVIEW® and Matlab® System Identification Toolbox

LabVIEW® for Windows™ 3.0.1 (Anon, 1993) and the Matlab® System Identification Toolbox 3.0 (Ljung, 1992) were used extensively throughout this thesis for experimental evaluation and open loop modelling, respectively. General comments on the value of this software are made on the basis of their use in this work with some experience in undergraduate course labs.

It was found that the initial learning curve of programming in LabVIEW was steep for both the author and undergraduate students although after several months the author became a somewhat proficient LabVIEW programmer. The graphical utilities and data acquisition capabilities of LabVIEW were found to be outstanding although the graphical language proved to be somewhat tedious for implementation of the AUDI and GPC algorithms.

The Matlab System Identification Toolbox was found to be an excellent tool for the batch identification and validation of industrial and experimental input-output data. The lack of a description in the documentation for the loss function used throughout the functions in the Identification Toolbox, while annoying, was resolved by analysis of the source code.

8.2 Recommendations

1. Base Predictive PID on Other Linear Control Structures

Utilizing the same procedure as outlined in Chapter 2, the use of other linear control structures can be used as a foundation for a model based PID control structure. Among the more interesting controllers is the very new unified approach of Saudagar *et al.* (1995; denoted as UPC) which extends GPC to output error noise model structures. The output error noise model, which allows a more accurate treatment of the noise dynamics, is more general than the equation error model found in GPC. It is likely that a predictive PID control law based on UPC will result in an improvement in performance for noisy plants compared to predictive PID based on GPC.

Because the infinite horizon predictive PID controller constants can be explicitly expressed in terms of the model parameters in a simple formulation, it is an excellent candidate for adaptive control when combined with a recursive identification algorithm. An intriguing topic worth considering is increasing the aggression of infinite horizon or mean level PID without significantly increasing its complexity. However, in Chapter 2 it was determined that using unrealistic values of the C_c polynomial to increase the aggression of infinite horizon PID were unsuccessful.

2. Practical Identification Issues

Although some theoretical concepts of identifiability have been introduced by Niu *et al.* (1994) and the concept of filtering the input-output data was formalized by Shook *et al.* (1991), practical identification remains very much an art form. Furthermore, numerous authors have suggested that the performance of an adaptive controller depends mainly on the quality of the plant model (Clarke, 1991; Fisher, 1991; Shook, 1991). It is imperative for the future success of adaptive control and recursive identification in industry that practical identification issues be addressed. General rules or guidelines regarding bad data, sufficient conditions for reliable estimation and model validation must be developed.

3. Simultaneous Time Delay and Parameter Estimation

Within the AUDI framework, it may be possible to simultaneously estimate the model parameters and the time delay for a fixed model order. Consider the following augmented regressor for a first order model and a possible range of time delay, $\hat{d} \in [d_{\min} \quad \cdots \quad d_{\max}]$

$$\begin{aligned}\phi(t) = [&-y(t-d_{\max}+d_{\min}-1) \quad u(t-d_{\max}-1) \quad -y(t-d_{\max}+d_{\min}) \quad \cdots \\ &\cdots \quad -y(t-2) \quad u(t-d_{\max}-1) \quad -y(t-1) \quad u(t-d_{\max}-1) \quad -y(t)]^T\end{aligned}$$

and the U matrix in a multidiagonal structure expressed as

$$U = \begin{bmatrix} 1 & \alpha_1^0 & \hat{\theta}_1^1 & & & & & \\ & 1 & \hat{\theta}_2^1 & \alpha_1^1 & & & 0 & \\ & & 1 & \alpha_2^1 & \hat{\theta}_1^2 & & & \\ & & & 1 & \hat{\theta}_2^2 & \ddots & & \\ & & & & 1 & \ddots & \alpha_1^n & \\ & & & & & \ddots & \alpha_2^n & \hat{\theta}_1^n \\ & 0 & & & & & 1 & \hat{\theta}_2^n \\ & & & & & & & 1 \end{bmatrix}$$

where n is $d_{\max} - d_{\min} + 1$. Readers are urged to consult Chapter 4 or Niu *et al*. (1992) for an explanation of the above notation. The solution with the lowest cost function will simultaneously solve for the first order model parameters and the time delay. However, the problem with this proposed method is that the U matrix that results from a batch or recursive solution places nonzero elements in the upper right positions. If these elements can be fixed to zero, then this method could work. Simultaneous estimation of model parameters and time delay would be very beneficial for adaptive control of processes with time varying dynamics and time varying delays. Since this problem was beyond the scope of this thesis, it is stated as a future recommendation.

4. Extend Steady State Weighting for $N_u > 1$

In Chapter 2 it was stated that for $N_u = 1$, increasing the steady state weight in GPC has the same effect as increasing N_2. For cases where $N_u > 1$, increasing the steady state weight has a different effect compared to increasing N_2. If the steady state property at $N_u = 1$ mentioned above could be extended to higher values of N_u, this would allow an infinite prediction horizon (by setting the finite weight to zero) for $N_u > 1$ which would result in a simple but very effective control law.

5. Multi Input Multi Output Predictive PID

Extension of the predictive PID approach to apply to a MIMO system would be a tremendous benefit because a MIMO PID algorithm currently does not exist. This may be possible by analyzing the linear formulation of MIMO GPC.

References

Anon, *LabVIEW for Windows User Manual*, National Instruments Corporation, Austin, TX, 1993.

Clarke, D.W., "Adaptive generalized predictive control," *Proc. Fourth Int. Conf. Chemical Process Control*, pp 395-417, 1991.

Elnaggar, A., G.A. Dumont and A.L Elshafei, "Delay estimation using variable regression," *Proc. American Control Conf.*, Vol. 3, pp 2812-2817, 1991.

Fisher, D.G., "Process control: an overview and personal perspective," *Can. J. Chem. Eng.*, **69,** pp 5-26, 1991.

Kwok, K., and S.L. Shah, "Long-range predictive control with a terminal matching condition," *Chemical Engineering Science*, Vol **49**, No. 9, pp. 1287-1300, 1994.

Ljung, L., *System Identification Toolbox for use with Matlab*, The MathWorks, Inc., Natick, Mass., 1992.

Niu, S., D.G. Fisher and D. Xiao, "An augmented UD identification algorithm," *Int. J. Control*, Vol. 56, No. 1, pp 193-211, 1992.

Saudagar, M.A. "Unified predictive control," internal report, Department of Chemical Engineering, University of Alberta, 1995.

Shook, D.S., "Identification issues in long range predictive control," Ph.D. thesis, University of Alberta, 1991.

Shook, D.S., C. Mohtadi and S.L. Shah, "A control relevant identification strategy for GPC," *IEEE Trans. Automatic Control*, Vol. 37, No. 7, pp. 975-980, 1992.